

МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)

На правах рукописи

УДК 519.254

Исаченко Роман Владимирович

ВЫБОР МОДЕЛИ ДЕКОДИРОВАНИЯ СИГНАЛОВ В ПРОСТРАНСТВАХ
ВЫСОКОЙ РАЗМЕРНОСТИ

05.13.17 — Теоретические основы информатики

Диссертация на соискание ученой степени
кандидата физико-математических наук

Научный руководитель:
д.ф.-м.н. В. В. Стрижов

Москва — 2021

Оглавление

	Стр.
Введение	5
Глава 1. Постановка задачи декодирования сигналов	12
1.1. Регрессионная модель в пространстве высокой размерности	12
1.2. Задача декодирования временных рядов	13
1.3. Линейная регрессия для многомерных сигналов	15
1.4. Методы снижения размерности пространства	16
Глава 2. Задача декодирования в пространстве высокой размерности	22
2.1. Постановка задачи декодирования	22
2.2. Доказательство корректности работы алгоритмов PLS и CCA	26
2.3. Алгоритмы PCA, PLS, CCA, DeepCCA в терминах задачи декодирования (работа Фаруха)	30
2.3.1. Линейная регрессия	30
2.3.2. Снижение размерности	31
2.3.3. Метод главных компонент	32
2.3.4. Метод частичных наименьших квадратов	33
2.3.5. Канонический анализ корреляций	34
2.3.6. Нелинейный канонический анализ корреляций	35
2.4. Вычислительный эксперимент	37
2.5. Вычислительный эксперимент из работы Фаруха	39
2.5.1. Анализ нелинейных зависимостей в задаче фильтрации шума	40
2.5.2. Анализ нелинейных моделей для восстановления изображений	43
Глава 3. Выбор признаков в задаче декодирования сигналов	46
3.1. Выбор признаков с помощью квадратичного программирования	47
3.2. Многоиндексный метод выбора признаков	48

3.2.1. Агрегация релевантностей целевых переменных	49
3.2.2. Симметричный учёт значимости признаков и целевых переменных	50
3.2.3. Минимаксная постановка задачи выбора признаков	51
3.2.4. Несимметричный учёт значимостей признаков и целевых переменных	54
3.3. Анализ методов учета значимостей целевых переменных	56
Глава 4. Выбор параметров нелинейных моделей с помощью квадратичного отбора признаков	
4.1. Задача выбора параметров для обучения моделей	63
4.2. Метод Ньютона решения задачи обучения параметров	64
4.3. Метод Ньютона с выбором параметром с помощью квадратичного программирования	67
4.4. Анализ значимостей параметров нелинейных моделей	69
Глава 5. Метрические методы в задаче анализа временных рядов	
5.1. Метрическое обучение в задачах кластеризации временных рядов	74
5.2. Алгоритм адаптивного метрического обучения	75
5.3. Постановка задачи метрического обучения с динамическим выравниванием временных рядов	78
5.3.1. Выравнивание временных рядов.	79
5.3.2. Метрическое обучение.	81
5.3.3. Классификация выравненных временных рядов в метрике Махаланобиса.	82
5.4. Анализ оптимального метрического пространства для задачи кластеризации	84
5.5. Анализ метрического пространства для задачи классификации временных рядов	87
Глава 6. Порождение признаков с помощью метамоделей	
6.1. Постановка задачи порождения признакового пространства	92

6.2. Модели порождение признакового пространства для временных рядов	93
6.3. Классификация временных рядов в порожденном признаковом про- странства	96
6.4. Анализ порожденных признаковых пространств	98
Заключение	104
Список основных обозначений	106
Список иллюстраций	107
Список таблиц	110
Список литературы	112

Введение

Диссертационная работа посвящена построению математических моделей машинного обучения в пространствах высокой размерности. Разработанные методы учитывают зависимости, имеющиеся в исходных данных, с целью построения простой и устойчивой модели.

Актуальность темы. В работе исследуется задача декодирования сигналов. Модель предсказывает отклик на входной исходный сигнал. При построении модели возникает задача построения признакового пространства.

Сложностью задачи является избыточность исходного описания данных. Исходное признаковое пространство является мультикоррелированным. При высокой мультикорреляции финальная предсказательная модель оказывается неустойчивой. Для построения простой, устойчивой модели применяются методы снижения размерности пространства [1, 2, 3] и выбора признаков [4, 5].

В работе рассматриваются задачи с векторной целевой переменной. При предсказании векторной целевой переменной анализируется структура целевого пространства. Предполагается, что целевое пространство имеет избыточную размерность. Предложены методы, которые учитывают зависимости как в исходном пространстве объектов, так и в пространстве целевой переменной. Предлагается отобразить пространства исходных и целевых сигналов в скрытые подпространства меньшей размерности. Для построения оптимальной модели предлагаются методы согласования скрытых пространств [6, 7, 8]. Предложенные методы позволяют учесть регрессионную компоненту между исходным и целевым сигналом, а также авторегрессионную компоненту целевого сигнала.

Методы снижения размерности пространства снижают размерность исходного пространства объектов, и, как следствие, сложность модели существенно снижается [9, 6, 10]. Алгоритмы снижения размерности находят оптимальные комбинации исходных признаков. Если число таких комбинаций существенно меньше, чем число исходных признаков, то полученное представление снижает

размерность. Цель снижения размерности – получение наиболее репрезентативных и информативных комбинаций признаков для решения задачи.

Выбор признаков является частным случаем снижения размерности пространства [4, 11]. Найденные комбинации признаков являются подмножеством исходных признаков. Таким образом отсеиваются шумовые неинформативные признаки. Рассматриваются два типа методов выбора признаков [5, 12, 13]. Первый тип методов не зависит от последующей предсказательной модели. Признаки отбираются на основе из свойств исходных пространств, а не на основе свойств модели. Второй тип методов отбирает признаки с учётом знания о предсказательной модели.

После нахождения оптимального представления данных с помощью снижения размерности, ставится задача нахождения оптимальной метрики в скрытом пространстве объектов [14, 15, 16, 17, 18]. В случае евклидова пространства естественным выбором метрики оказывается квадратичная норма. Задача метрического обучения заключается в нахождении оптимальной метрики, связывающей объекты.

В качестве прикладной задачи анализируется задача построения нейрокомпьютерного интерфейса [19, 20]. Цель состоит в извлечении информации из сигналов мозговой активности [21, 22, 23]. В качестве исходных сигналов может выступать электроэнцефалограмма или электрокортикограмма. Примером целевого сигнала может быть движение конечностей индивидуума. Задача модели построить адекватную и эффективную модель преобразования исходного сигнала в целевой сигнал. Пространство частотных характеристик мозговых сигналов является чрезвычайно избыточным [24, 25]. Пространство целевого сигнала имеет выраженную авторегрессионную временную компоненту. Построение модели без учёта имеющихся зависимостей приводит к неустойчивости модели.

В диссертации решается задача декодирования с векторной целевой переменной. Для построения оптимальной модели декодирования временных рядов предлагаются методы проекции в скрытое пространство. Исходные и целевые

сигналы проецируются в пространство существенно меньшей размерности. Для связи проекций исходного и целевого сигнала предлагаются методы согласования. Рассматриваются многомодальные данные, природа источников измерений различны. Рассматриваются как линейные методы декодирования, так и их нелинейные обобщения. Доказаны теоремы об оптимальности предложенных методов снижения размерности пространства, и методов выбора признаков.

Цели работы.

1. Исследовать свойства решения задачи декодирования сигналов с многомерной целевой переменной.
2. Предложить способ снижения размерности пространства, учитывающий зависимости как в исходном пространстве сигналов, так и в целевом пространстве.
3. Предложить процедуру выбора признаков для задачи декодирования сигналов.
4. Исследовать свойства линейных и нелинейных моделей для решения поставленной модели. Получить теоретические оценки оптимальности моделей.
5. Провести вычислительный эксперимент для проверки адекватности предложенных методов.

Основные положения, выносимые на защиту.

1. Предложен метод снижения размерности пространства, отображающий независимую и целевую переменные в единое скрытое низкоразмерное представление.
2. Предложены методы выбора признаков для задач с многомерной целевой переменной, учитывающие структуры пространств.
3. Доказаны теоремы для симметричного и несимметричного учета значимости признаков, а также для минимаксной постановки задачи выбора признаков.

4. Предложен алгоритм выбора наиболее релевантных параметров для оптимизации нелинейной модели. Исследованы свойства алгоритма.
5. Предложен алгоритм метрического обучения для временных рядов с процедурой их выравнивания.
6. Разработан программный комплекс, включающий прогностические модели данных высокой размерности. Проведены вычислительные эксперименты, подтверждающие адекватность моделей.

Методы исследования. Для достижения поставленных целей используются линейные и нелинейные алгоритмы регрессии. Для анализа временных рядов используются классические авторегрессионные методы. Для извлечения признаков используются частотные характеристики временного ряда. Для построения скрытого пространства используются линейные методы снижения размерности пространства, их нелинейные модификации, а также нейросетевые методы. Для выбора признаков наряду с классическими методами, используются методы, основанные на решении задачи квадратичного программирования. Для построения метрического пространства используются методы условной выпуклой оптимизации.

Научная новизна. Предложены методы построения моделей декодирования временных рядов, учитывающие структуры пространств исходных и целевых сигналов. Предложены методы проекции сигналов в скрытое пространство, а также процедуры согласования образов. Предложены алгоритмы выбора признаков с помощью квадратичного программирования. Предложен метод выбора параметров для оптимизации с помощью выбора признаков. Предложены алгоритмы построения оптимального метрического пространства для задачи анализа временных рядов.

Теоретическая значимость. Доказаны теоремы об оптимальности предлагаемых моделей декодирования временных рядов. Доказаны теоремы о кор-

ректности рассматриваемых моделей проекций в скрытое пространство. Доказаны теоремы о достижении точки равновесия для предлагаемых алгоритмов выбора признаков.

Практическая значимость. Предложенные в работе методы предназначены для декодирования множества временных рядов сигналов электрокортикограмм, а также нестационарных временных рядов; выбора оптимальных частотных характеристик сигналов; выбора наиболее информативных параметров модели; классификации и кластеризации временных рядов физической активности.

Степень достоверности и апробация работы. Достоверность результатов подтверждена математическими доказательствами, экспериментальной проверкой результатов предлагаемых алгоритмов на реальных данных, публикациями результатов в рецензируемых научных изданиях, в том числе рекомендованных ВАК. Результаты работы докладывались и обсуждались на следующих научных конференциях.

1. Р. В. Исаченко. Метрическое обучение в задачах мультиклассовой классификации временных рядов. *Международная научная конференция «Ломоносов»*, 2016, [26].
2. R. G. Neychev, A. P. Motrenko, R. V. Isachenko, A. S. Inyakin, and V. V. Strijov. Multimodel forecasting multiscale time series in internet of things. *Международная научная конференция «11th International Conference on Intelligent Data Processing: Theory and Applications»*, 2016, [27].
3. Р. В. Исаченко, И. Н. Жариков, и А. М. Бочкарев. Локальные модели для классификации объектов сложной структуры. *Всероссийская научная конференция «Математические методы распознавания образов»*, 2017, [28].
4. R. V. Isachenko and V. V. Strijov. Dimensionality reduction for multicorrelated signal decoding with projections to latent space. *Международная научная конференция «Математические методы распознавания образов»*, 2017, [29].

родная научная конференция «12th International Conference on Intelligent Data Processing: Theory and Applications», 2018, [29].

5. Р. В. Исаченко, В. В. Стрижов. Снижение размерности в задаче декодирования временных рядов. *Международная научная конференция «13th International Conference on Intelligent Data Processing: Theory and Applications»*, 2020, [30].

Работа поддержана грантами Российского фонда фундаментальных исследований.

1. 19-07-00885, Российский фонд фундаментальных исследований в рамках гранта «Выбор моделей в задачах декодирования временных рядов высокой размерности».
2. 16-37-00485, Российский фонд фундаментальных исследований в рамках гранта «Развитие методов выбора признаков в условиях мультиколлинеарности».
3. 16-07-01160, Российский фонд фундаментальных исследований в рамках гранта «Развитие теории обучения по предпочтениям с использованием частично упорядоченных множеств экспертов оценок».
4. 16-07-01154, Российский фонд фундаментальных исследований в рамках гранта «Новые методы прогнозирования на базе субквадратичного анализа метрических конфигураций».

Публикации по теме диссертации. Основные результаты по теме диссертации изложены в 6 печатных изданиях, 4 из которых изданы в журналах, рекомендованных ВАК.

1. Исаченко Р. В., Катруца А. М. Метрическое обучение и снижение размерности пространства в задачах кластеризации // Машинное обучение и анализ данных, 2016. Т. 2. № 1. С. 17–25 [31].
2. Исаченко Р. В., Стрижов В. В. Метрическое обучение в задачах мультиклассовой классификации временных рядов // Информатика и её применения, 2016. Т. 10. № 2. С. 48–57 [32].

3. Isachenko R. et al. Feature Generation for Physical Activity Classification // Artificial Intelligence and Decision Making, 2018. № 3. C. 20–27 [33].
4. Isachenko R. V., Strijov V. V. Quadratic programming optimization with feature selection for nonlinear models // Lobachevskii Journal of Mathematics, 2018. Т. 39. № 9. С. 1179–1187 [34].
5. Isachenko R. V., Vladimirova M. R., Strijov V. V. Dimensionality Reduction for Time Series Decoding and Forecasting Problems // DEStech Transactions on Computer Science and Engineering, 2018. №. optim. [35].
6. Исаченко Р.В., Яушев Ф.Ю., Стрижов В.В. Модели согласования скрытого пространства в задаче прогнозирования // Системы и средства информатики, 2021. Т. 31 № 1. [36]
7. **expert systems**

Структура и объем работы.

Личный вклад. Все приведенные результаты, кроме отдельно оговоренных случаев, получены диссертантом лично при научном руководстве д.ф.-м.н. В. В. Стрижова.

Краткое содержание работы по главам.

Глава 1

Постановка задачи декодирования сигналов

В данной главе ставится общая задача декодирования временных рядов. Приводится обзор стандартных методов анализа временных рядов. Ставится задача построения оптимальной линейной авторегрессионной модели декодирования. Приведен обзор широко используемых методов снижения размерности пространства, их обобщений и модификаций.

1.1. Регрессионная модель в пространстве высокой размерности

Задача декодирования формулируется следующим образом. Требуется построить регрессионную модель между объектами из двух пространств: \mathbb{X} и \mathbb{Y} . Пространства обладают высокой, избыточной размерностью. Регрессионная модель оказывается неустойчивой. Для построения простой, точной и устойчивой модели предлагается учесть наличие внутренней структуры обоих пространств. Применяются методы снижения размерности пространств для независимой и целевой переменной. Итоговая регрессионная модель строится путём согласования образов исходных объектов в низкоразмерном пространстве.

Формализуем описанную задачу. Пусть $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^n$ – независимая переменная, $\mathbf{y} \in \mathbb{Y} \subset \mathbb{R}^r$ – целевая переменная.

Определение 1. Назовём *примером* пару (\mathbf{x}, \mathbf{y}) , состоящую из реализации независимой переменной $\mathbf{x} \in \mathbb{X}$ и целевой переменной $\mathbf{y} \in \mathbb{Y}$.

Определение 2. *Выборкой* $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$ будем называть заданное множество примеров $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$. Здесь $\mathbf{X} \in \mathbb{R}^{m \times n}$ – матрица независимой переменной, $\mathbf{Y} \in \mathbb{R}^{n \times k}$ – матрица целевой переменной:

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]^T = [\boldsymbol{\chi}_1, \dots, \boldsymbol{\chi}_n]; \quad \mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_m]^T = [\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_r].$$

Столбцы $\boldsymbol{\chi}_j, j = 1, \dots, n$ матрицы \mathbf{X} являются признаками объекта, столбцы $\boldsymbol{\nu}_j, j = 1, \dots, r$ матрицы \mathbf{Y} являются целевыми столбцами.

Определение 3. Назовём *моделью* отображение $f : \mathbb{X} \rightarrow \mathbb{Y}$ из пространства независимой переменной в пространство целевой переменной.

Задача восстановления регрессии состоит в нахождении оптимальной модели f^* по известной выборке \mathcal{D} . Под оптимальностью понимается нахождение такой модели, которая бы доставляла минимум некоторой функции ошибки

$$f^* = \arg \min_f \mathcal{L}(f, \mathbf{X}, \mathbf{Y}). \quad (1.1)$$

Задача поиска оптимальной модели является задачей функциональной оптимизации. Для сужения пространства поиска моделей будем рассматривать параметрические модели $f(\mathbf{x}, \Theta)$, где Θ являются *параметрами модели*. Тогда задача (1.1) сводится к задаче поиска набора оптимальных параметров

$$\Theta^* = \arg \min_{\Theta} \mathcal{L}(\Theta, \mathbf{X}, \mathbf{Y}). \quad (1.2)$$

В данной диссертации рассматривается случай, когда размерность пространств \mathbb{X}, \mathbb{Y} является избыточной. В таком случае решение задачи (1.2) оказывается неустойчивым. Рассмотрим в качестве примера задачу восстановления линейной регрессии.

1.2. Задача декодирования временных рядов

Рассмотрим задачу декодирования временных рядов.

Определение 4. Пусть $\mathbf{S} = [\mathbf{s}_1, \dots, \mathbf{s}_T]$ – *сегмент многомерного временного ряда*. Вектор \mathbf{s}_t содержит значения временного ряда в момент времени t . В случае одномерного временного ряда \mathbf{s}_t является скаляром.

Определение 5. Назовём *задачей декодирования временных рядов* задачу восстановления значений сегментов целевых временных рядов \mathbf{S}_y по сегментам наблюдаемых независимых временных рядов \mathbf{S}_x .

Если сегмент целевого временного ряда \mathbf{S}_y является продолжением наблюдаемого сегмента \mathbf{S}_x задача декодирования сводится к построению прогности-

ческом модели временного ряда. Для построения модели прогнозирования временных рядов широко используются два класса линейных методов: авторегрессионные модели [37, 38] и модели скользящего среднего [37, 38]. Авторегрессионные модели AR(p) строят прогноз в виде линейной комбинации p предыдущих значений временного ряда. Модели скользящего среднего MA(q) вместо предыдущих значений временного ряда используют комбинацию ошибок. Модель ARMA(p, q) [39] является комбинацией двух описанных подходов. ARMA(p, q) задает модель как линейную комбинацию p предыдущих значений временного ряда и q предыдущих значений ошибок. Для нахождения оптимальных параметров p и q модели ARMA используются автокорреляционная и частная автокорреляционная функции.

Модель ARMA используется для стационарных временных рядов, отвечающим строгим статистическим предположениям. На практике встречается огромное количество нестационарных временных рядов подверженных тренду, сезонности или цикличности. Модель ARIMA(p, d, q) [39] обобщает модель ARMA для случая нестационарных временных рядов. ARIMA берёт разности порядка d от исходного временного ряда для достижения стационарности данных. При этом на практике оказывается достаточным положить $d = 1$. Заметим, что при $d = 0$ модель ARIMA эквивалентна модели ARMA. Полезным обобщение модели ARIMA является модель AFRIMA [40]. Модель позволяет задать параметр d в виде вещественного числа.

Модель ARIMA плохо справляется с сезонными временными рядами. В работе [37] была предложена модель SARIMA, которая вводит в модель учет сезонной компоненты.

Задача декодирования временных рядов декомпозируется на следующие подзадачи.

- Порождение признакового пространства. Данный этап включает в себя процедуру извлечения признаков из исходных значений сигналов. Процедура порождения признакового пространства может быть основана на

экспертных знания или же являться моделью машинного обучения. Данная подзадача подробно рассмотрена в главе 6.

- Снижение размерности пространства или выбор признаков. Исходные временные ряды, а также порожденное признаковое пространство оказывается избыточным, что приводит к избыточности и неустойчивости модели. Алгоритмы снижения размерности и выбора признаков подробно изложены в главах 2, 3.
- Построение модели. После нахождения оптимального низкоразмерного представления исходных данных ставится задача выбора оптимальной модели декодирования.

1.3. Линейная регрессия для многомерных сигналов

Предполагается, что между объектами \mathbf{x} и ответами \mathbf{y} существует зависимость вида

$$\mathbf{y} = f(\mathbf{x}, \Theta) + \boldsymbol{\varepsilon}, \quad (1.3)$$

где f – параметрическая модель регрессионной зависимости, Θ – пространство параметров модели, $\boldsymbol{\varepsilon} \in \mathbb{R}^m$ – вектор регрессивных остатков. Необходимо восстановить зависимость f по заданной наблюдаемой выборке \mathcal{D} .

Предположим, что зависимость $f(\mathbf{x}, \Theta)$ линейная:

$$\mathbf{y} = f(\mathbf{x}, \Theta) + \boldsymbol{\varepsilon} = \Theta \mathbf{x} + \boldsymbol{\varepsilon}, \quad (1.4)$$

где $\Theta \in \mathbb{R}^{r \times n}$ – матрица параметров модели.

Необходимо найти матрицу параметров модели Θ при известной выборке $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$. Оптимальные параметры Θ определяются минимизацией функции ошибки $\mathcal{L}(\Theta, \mathbf{X}, \mathbf{Y})$. При решении задачи линейной регрессии в качестве такой функции ошибки рассматривается квадратичная функция потерь:

$$\mathcal{L}(\Theta, \mathbf{X}, \mathbf{Y}) = \left\| \mathbf{Y}_{m \times r} - \mathbf{X}_{m \times n} \cdot \Theta_{n \times r}^\top \right\|_2^2 \rightarrow \min_{\Theta}. \quad (1.5)$$

Решением (1.5) является следующая матрица:

$$\Theta = \mathbf{Y}^\top \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1}.$$

Наличие линейной зависимости между столбцами матрицы \mathbf{X} приводит к неустойчивому решению задачи оптимизации (1.5). Если существует вектор $\boldsymbol{\alpha} \neq \mathbf{0}_n$ такой, что $\mathbf{X}\boldsymbol{\alpha} = \mathbf{0}_m$, то добавление $\boldsymbol{\alpha}$ к любому столбцу матрицы Θ не меняет значение функции потерь $\mathcal{L}(\Theta, \mathbf{X}, \mathbf{Y})$. В этом случае матрица $\mathbf{X}^\top \mathbf{X}$ близка к сингулярной и не обратима. Чтобы избежать сильной линейной зависимости между признаками, в данной работе исследуются методы снижения размерности и выбора признаков.

1.4. Методы снижения размерности пространства

2 предложения

Метод главных компонент для задачи декодирования. Для устранения линейной зависимости и снижения размерности входного пространства объектов широко используется метод главных компонент (principal component analysis, PCA). Метод PCA находит низкоразмерное представление матрицы $\mathbf{X} = \mathbf{T}\mathbf{P}^\top$, такое что новое представление $\mathbf{T} \in \mathbb{R}^{m \times l}$ содержит максимальную долю дисперсии исходной матрицы. При этом матрица отображения $\mathbf{P} \in \mathbb{R}^{n \times l}$ является ортогональной ($\mathbf{P}^\top \mathbf{P} = \mathbf{I}$) и содержит правые собственные вектора матрицы ковариаций $\mathbf{X}^\top \mathbf{X}$.

Метод PCA является базовым методом снижения размерности пространства. Существует множество модификаций базового метода. Вероятностный PCA [9] рассматривает задачу снижения размерности в терминах вероятностной модели, решая задачу с помощью вариационного EM алгоритма. Разреженный PCA [41] вводит в постановку задачи lasso регуляризацию для того, чтобы сделать матрицу отображения \mathbf{P} разреженной и более интерпретируемой. Нелинейный ядерный PCA [42] отображает исходные данные с помощью

нелинейного отображения и использует RKHS для решения исходной задачи.

После нахождения матрицы отображения \mathbf{P} задача (1.5) принимает вид

$$\mathcal{L}(\mathbf{B}, \mathbf{T}, \mathbf{Y}) = \left\| \mathbf{Y}_{m \times r} - \mathbf{T}_{m \times l} \cdot \mathbf{B}_{r \times l}^T \right\|_2^2 \rightarrow \min_{\mathbf{B}}. \quad (1.6)$$

Модель прогнозирования (1.4) в случае снижения размерности с помощью PCA принимает вид:

$$\mathbf{y} = \mathbf{B}\mathbf{t} + \boldsymbol{\varepsilon} = \mathbf{B}\mathbf{P}\mathbf{x} + \boldsymbol{\varepsilon} = \boldsymbol{\Theta}\mathbf{x} + \boldsymbol{\varepsilon}, \text{ где } \boldsymbol{\Theta} = \mathbf{B}\mathbf{P}. \quad (1.7)$$

Метод частичных наименьших квадратов для задачи декодирования. Основным недостатком метода PCA является отсутствие учёта взаимосвязи между признаками \mathbf{x}_j и целевыми векторами $\boldsymbol{\nu}_j$. Алгоритм частичных наименьших квадратов проецирует матрицу объектов \mathbf{X} и матрицу ответов \mathbf{Y} в скрытое пространство малой размерностью l ($l < n$). Алгоритм PLS находит в скрытом пространстве матрицы $\mathbf{T}, \mathbf{U} \in \mathbb{R}^{m \times l}$, которые лучше всего описывают оригинальные матрицы \mathbf{X} и \mathbf{Y} . При этом PLS максимизирует ковариацию между столбцами \mathbf{t} и \mathbf{u} матриц \mathbf{T} и \mathbf{U} соответственно.

Алгоритм PLS был впервые предложен в работах [6, 43, 44]. Подробное описание алгоритма приведено в работах [45, 46, 47, 48, 49]. В работах [7, 50] приведен обзор обобщений базовой модели PLS. В работе [2] приведена модификация алгоритма PLS для получения разреженного набора признаков.

Матрица объектов \mathbf{X} и целевая матрица \mathbf{Y} проецируются на латентное пространство следующим образом:

$$\mathbf{X}_{m \times n} = \mathbf{T}_{m \times l} \cdot \mathbf{P}_{l \times n}^T + \mathbf{F}_{m \times n} = \sum_{k=1}^l \mathbf{t}_k \cdot \mathbf{p}_k^T + \mathbf{F}_{m \times n}, \quad (1.8)$$

$$\mathbf{Y}_{m \times r} = \mathbf{U}_{m \times l} \cdot \mathbf{Q}_{l \times r}^T + \mathbf{E}_{m \times r} = \sum_{k=1}^l \mathbf{u}_k \cdot \mathbf{q}_k^T + \mathbf{E}_{m \times r}. \quad (1.9)$$

Здесь \mathbf{T} и \mathbf{U} – образы исходных матриц в скрытом пространстве, причём столбцы матрицы \mathbf{T} ортогональны; \mathbf{P} и \mathbf{Q} – матрицы перехода; \mathbf{E} и \mathbf{F} – матрицы остатков. Алгоритм PLS максимизирует линейную зависимость между столбцами матриц \mathbf{T} и \mathbf{U}

$$\mathbf{U} \approx \mathbf{T}\mathbf{B}, \quad \mathbf{B} = \text{diag}(\beta_k), \quad \beta_k = \mathbf{u}_k^\top \mathbf{t}_k / (\mathbf{t}_k^\top \mathbf{t}_k)$$

Алгоритм решает следующую оптимизационную задачу:

$$\max_{\|\mathbf{p}\|_2 = \|\mathbf{q}\|_2 = 1} [\text{cov}(\mathbf{X}\mathbf{p}, \mathbf{Y}\mathbf{q})^2] = \max_{\mathbf{p}, \mathbf{q}} \frac{\mathbf{p}^\top \mathbf{X}^\top \mathbf{Y}\mathbf{q}}{\sqrt{\mathbf{p}^\top \mathbf{p}} \sqrt{\mathbf{q}^\top \mathbf{q}}}. \quad (1.10)$$

Детальное описание алгоритма PLS с доказательством его корректности приведено в ??.

Для демонстрации разницы между алгоритмами PCA, PLS был проведен модельный эксперимент для случая, когда размерности пространств объектов, ответов и латентного пространства равны 2 ($n = r = l = 2$). На Рис. 1.1 показаны результаты работы методов. Синими и зелёными точками изображены объекты \mathbf{x}_i и целевые переменные \mathbf{y}_i . Точки \mathbf{X} сгенерированы из нормального распределения с нулевым матожиданием. Точки \mathbf{Y} линейным образом зависят от второй главной компоненты pc_2 матрицы \mathbf{X} и не зависят от первой главной компоненты pc_1 . Красным контуром показаны линии уровня матриц ковариаций распределений. Чёрным изображены единичные окружности. Красные стрелки соответствуют главным компонентам матриц \mathbf{X} и \mathbf{Y} . Чёрные стрелки соответствуют векторам матриц \mathbf{W} и \mathbf{C} алгоритма PLS. Данные матрицы содержат вектора, являющиеся аналогами главных компонент алгоритма PCA. Учёт взаимной связи между матрицами \mathbf{X} и \mathbf{Y} отклоняет вектора \mathbf{w}_k и \mathbf{c}_k от направления главных компонент.

При снижении размерности пространств до одного признака алгоритм PCA выберет первую главную компоненту pc_1 , отбросив компоненту pc_2 , так как первая компонента объясняет большую части дисперсии исходной матрицы \mathbf{X} . При этом матрица \mathbf{Y} не зависит от pc_1 . Тем самым финальная модель окажется

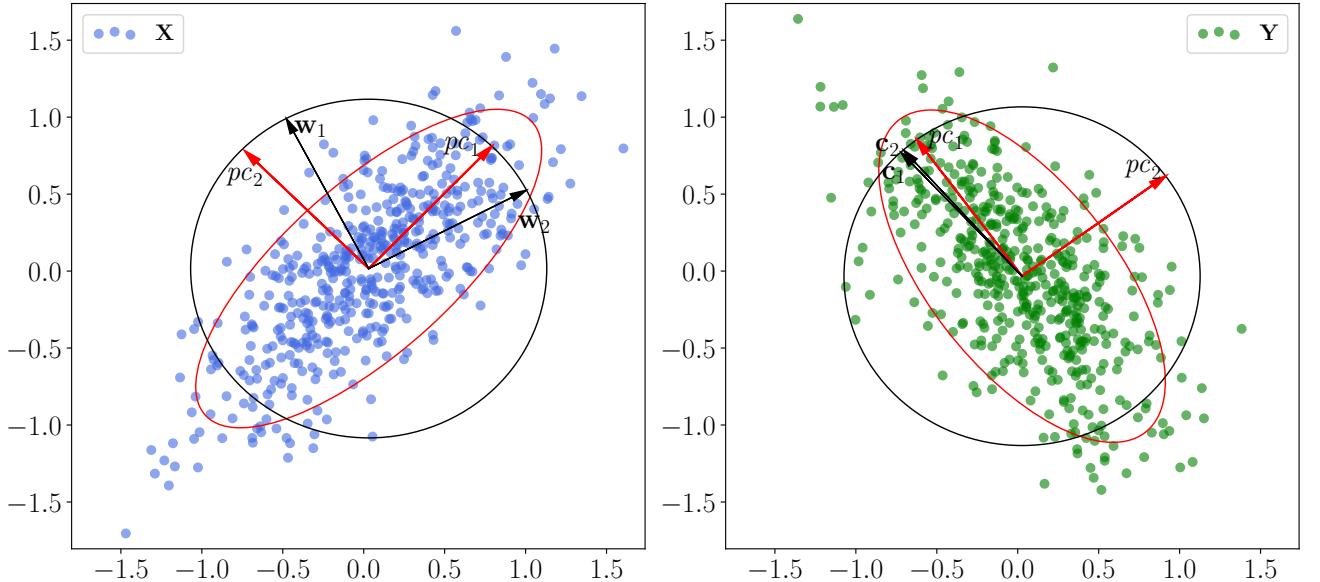


Рис. 1.1: Модельный пример работы алгоритмов PCA и PLS

не оптимальной. Алгоритм PLS позволяет побороться с данной проблемой.

Канонический анализ корреляций для задачи декодирования. Канонический корреляционный анализ (canonical correlation analysis, CCA) широко применяется для поиска взаимосвязи между двумя наборами переменных [10, 51]. Оптимизационная задача CCA похожа на оптимизационную задачу PLS (1.10) с той лишь разницей, что вместо максимизации ковариации CCA максимизирует корреляцию:

$$\max_{\|\mathbf{p}\|_2 = \|\mathbf{q}\|_2 = 1} [\text{corr}(\mathbf{X}\mathbf{p}, \mathbf{Y}\mathbf{q})^2] = \max_{\mathbf{p}, \mathbf{q}} \frac{\mathbf{p}^\top \mathbf{X}^\top \mathbf{Y} \mathbf{q}}{\sqrt{\mathbf{p}^\top \mathbf{X}^\top \mathbf{X} \mathbf{p}} \sqrt{\mathbf{q}^\top \mathbf{Y}^\top \mathbf{Y} \mathbf{q}}}. \quad (1.11)$$

На Рис. 1.2 показан результат работы алгоритма. Основное различие состоит в том, что вектора \mathbf{c}_1 и \mathbf{c}_2 в данном случае становятся ортогональными.

В Таблице 1.1 приведены значения квадратичной ошибки $\mathcal{L}(\Theta, \mathbf{X}, \mathbf{Y})$ для алгоритмов линейной регрессии, PCA и PLS. Линейная регрессия отлично справляется с данной задачей. Ошибка алгоритма PCA наибольшая, что подтверждает факт, что для данной ситуации алгоритм не находит нужных зависимостей в пространстве целевой переменной. Алгоритмы PLS и CCA показывают схожие результаты.

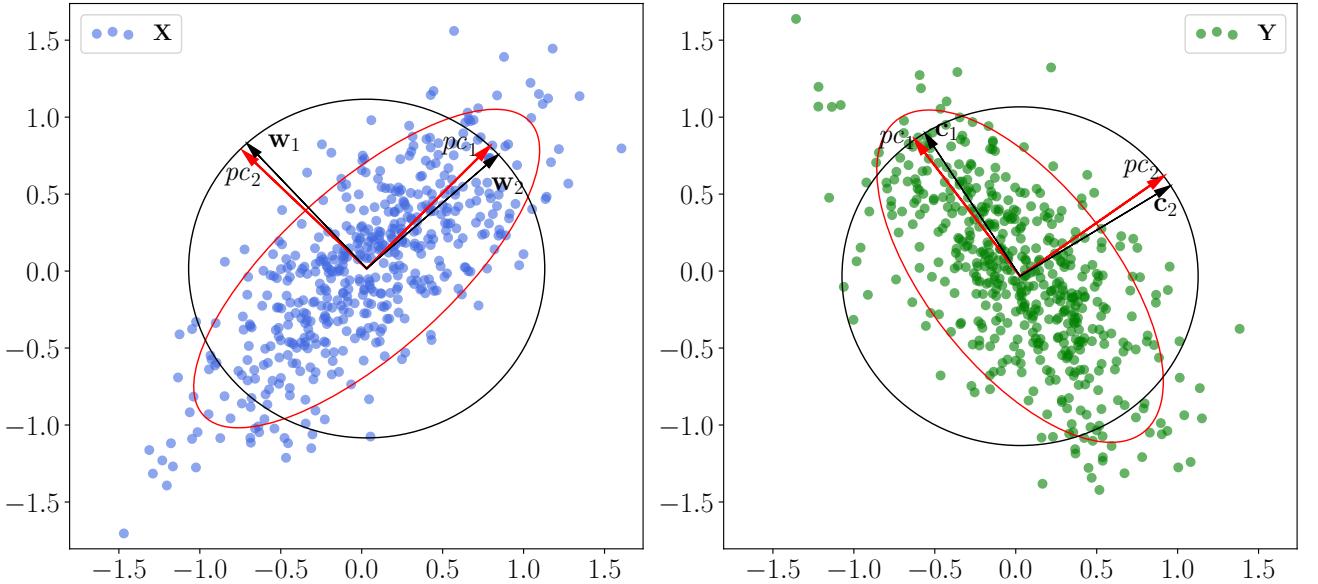


Рис. 1.2: Модельный пример работы алгоритмов PCA и CCA

Линейная регрессия	PCA	PLS	CCA
0.01	0.24	0.13	0.13

Таблица 1.1: Средняя квадратичная ошибка на модельном примере для алгоритмов линейной регрессии, PCA, PLS, CCA

Нелинейный ядерный CCA [52, 53, 54, 55] является обобщением базового метода. CCA и ядерный CCA широко используются для задач обучения без учителя [56, 57]. Метод имеет область применения от анализа хемометрических [58] и биологических [59] данных до обработки естественного языка [60, 61], аудиосигналов [62, 63] и компьютерного зрения [64].

В работе [65] был впервые предложено обобщение алгоритма CCA, работающего с нейросетями. Предложенный алгоритм DeepCCA максимизирует корреляцию между представлениями, полученными на выходе нейросети:

$$\begin{aligned} \max_{\|\mathbf{p}\|_2 = \|\mathbf{q}\|_2 = 1} & [\text{corr}(\mathbf{f}(\mathbf{X}, \mathbf{W}_x) \cdot \mathbf{p}, \mathbf{g}(\mathbf{Y}, \mathbf{W}_y) \cdot \mathbf{q})^2] = \\ & = \max_{\mathbf{p}, \mathbf{q}} \frac{\mathbf{p}^\top \mathbf{f}(\mathbf{X}, \mathbf{W}_x)^\top \mathbf{g}(\mathbf{Y}, \mathbf{W}_y) \mathbf{q}}{\sqrt{\mathbf{p}^\top \mathbf{f}(\mathbf{X}, \mathbf{W}_x)^\top \mathbf{f}(\mathbf{X}, \mathbf{W}_x) \mathbf{p}} \sqrt{\mathbf{q}^\top \mathbf{g}(\mathbf{Y}, \mathbf{W}_y)^\top \mathbf{g}(\mathbf{Y}, \mathbf{W}_y) \mathbf{q}}} \quad (1.12) \end{aligned}$$

Здесь $\mathbf{f}(\mathbf{X}, \mathbf{W}_x)$ и $\mathbf{g}(\mathbf{Y}, \mathbf{W}_y)$ – нелинейные проекции исходных и целевых объектов. В статье [66] приведен обширный обзор модификаций нейросетевого ССА для работы с многовидовыми данными. Главным недостатком нейросетевого ССА является вычислительная сложность. В работе [67] предложена релаксация исходного лосса, которая способна масштабироваться под работу с большими глубокими моделями.

Тензорные линейные методы для задачи декодирования.

Если исходный объект \mathbf{x} является вектором, а тензором более высокого порядка, то для построения модели тензор может быть вытянут в вектор [68]. В таком случае модель не учитывает имеющиеся зависимости между различными направлениями исходного тензора. Для учета таких зависимостей используются тензорные версии алгоритма PLS [69, 24, 70].

Многомодальные данные в задаче декодирования. Исходный объект может иметь несколько модальностей. Примерами таких модальностей могут быть выровненные аудио и видео [71, 72], аудио и артикуляция [73], изображение и текстовая аннотация [55, 74, 75], параллельный корпус текстов [57, 60, 76, 77].

В случае если для каждого объекта имеется более двух модальностей, то для построения скрытого пространства для каждого из них применяются два класса подходов. Первый подход состоит в построении скрытого пространства для каждой пары модальностей объекта [78, 79]. Второй же подход состоит в построении общего единого скрытого пространства для всех модальностей [80, 81].

Глава 2

Задача декодирования в пространстве высокой размерности

В данной главе ставится формальная постановка задачи декодирования в терминах проекций в скрытое пространство. Вводятся понятия скрытого пространства, и процедура согласования образов. Доказываются теоремы об оптимальном выборе модели декодирования.

2.1. Постановка задачи декодирования

Пусть имеется два множества временных рядов $\mathcal{S}^x = \{\mathbf{s}_i^x\}_{i=1}^m$ и $\mathcal{S}^y = \{\mathbf{s}_i^y\}_{i=1}^r$, состоящие из m и r временных рядов соответственно. Первое множество \mathcal{S}^x является множеством временных рядов m входных сигналов. Второе множество \mathcal{S}^y является множеством временных рядов r целевых сигналов. Каждый временной ряд $\mathbf{s} = (s_1, s_2, \dots, s_T)$ является последовательность измерений некоторой величины в течение времени.

Определение 6. Временное представление $\mathbf{x}_t = ([\mathbf{s}_1^x]_t, \dots, [\mathbf{s}_m^x]_t) \in \mathbb{R}^m$ состоит из измерений временных рядов входных сигналов в момент времени t . Аналогично временное представление $\mathbf{y}_t = ([\mathbf{s}_1^y]_t, \dots, [\mathbf{s}_r^y]_t) \in \mathbb{R}^r$ состоит из измерений временных рядов целевых сигналов в момент времени t .

Определение 7. Определим представление предыстории длины h для момента времени t множества временных рядов входных сигналов \mathcal{S}^x как совокупность представлений $\mathbf{X}_{t,h} = [\mathbf{x}_{t-h+1}, \dots, \mathbf{x}_t]^T \in \mathbb{R}^{h \times m}$. Аналогично определим представление предыстории длины h для момента времени t множества временных рядов целевых сигналов \mathcal{S}^y как совокупность представлений $\mathbf{Y}_{t,h} = [\mathbf{y}_{t-h+1}, \dots, \mathbf{y}_t]^T \in \mathbb{R}^{h \times r}$.

Определение 8. Определим представление горизонта прогнозирования длины p для момента времени t множества временных рядов входных сигналов

\mathcal{S}^x как совокупность представлений $\mathbf{X}_{t,p} = [\mathbf{x}_{t+1}, \dots, \mathbf{x}_{t+p}]^\top \in \mathbb{R}^{p \times m}$. Аналогично определим представление горизонта прогнозирования длины p для момента времени t множества временных рядов целевых сигналов \mathcal{S}^y как совокупность представлений $\mathbf{Y}_{t,r} = [\mathbf{y}_{t+1}, \dots, \mathbf{y}_{t+r}]^\top \in \mathbb{R}^{p \times r}$.

Задача авторегрессионного декодирования состоит в построении предсказательной модели f_{AR} , дающий прогноз представления горизонта прогнозирования множества временных рядов по представлению предыстории прогнозирования того же множества временных рядов.

Определение 9. Предсказательная модель $f_{\text{AR}}^x : \mathbb{R}^{h \times m} \rightarrow \mathbb{R}^{p \times m}$ является авторегрессионной моделью, которая по представлению предыстории $\mathbf{X}_{t,h}$ множества временных рядов входных сигналов \mathcal{S}^x предсказывает представление горизонта прогнозирования $\mathbf{X}_{t,p}$ множества временных рядов входных сигналов \mathcal{S}^x . Аналогично вводится предсказательная модель $f_{\text{AR}}^y : \mathbb{R}^{h \times r} \rightarrow \mathbb{R}^{p \times r}$ для множества целевых сигналов \mathcal{S}^y .

Суть авторегрессионного декодирования заключается в предсказании будущего прогноза временного ряда по его же предыстории.

Определение 10. Определим задачу регрессионного декодирования как задачу построения предсказательной модели $f_{\text{R}}^{xy} : \mathbb{R}^{h \times m} \rightarrow \mathbb{R}^{p \times r}$, которая по представлению предыстории $\mathbf{X}_{t,h}$ множества временных рядов входных сигналов \mathcal{S}^x предсказывает представление горизонта прогнозирования $\mathbf{Y}_{t,p}$ множества временных рядов целевых сигналов \mathcal{S}^y .

Отличие регрессионного декодирования от авторегрессионного декодирования состоит в том, что в случае регрессионного декодирования представление предыстории и представление горизонта прогнозирования получены из временных рядов разных пространств, а именно предыстория получена из множества входных сигналов, в то время как горизонт прогнозирования получен из множества целевых сигналов.

Определение 11. Общая задача декодирования состоит в построении предсказательной модели $f^{xy} : \mathbb{R}^{h_x \times m} \times \mathbb{R}^{h_y \times r} \rightarrow \mathbb{R}^{p \times r}$, которая по представлениям

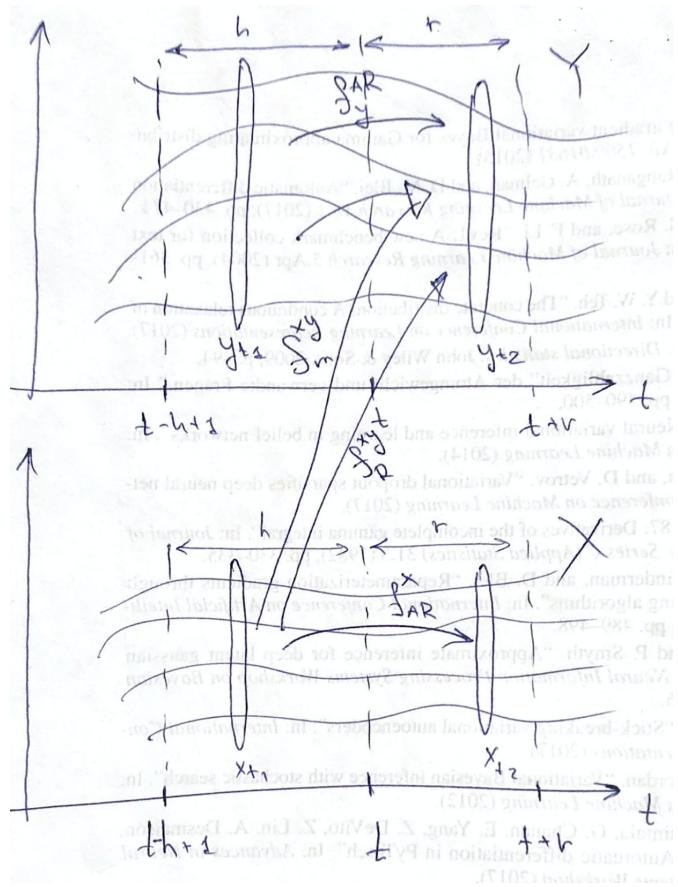


Рис. 2.1: Схема построения моделей декодирования

предыстории \mathbf{X}_{t,h_x} и \mathbf{Y}_{t,h_y} временных рядов входных и целевых сигналов предсказывает представление горизонта прогнозирования $\mathbf{Y}_{t,r}$ временных рядов целевых сигналов.

Отметим, что авторегрессионная модель f_{AR}^y и регрессионная модель f_R^{xy} являются частными случаями общей задачи декодирования. А именно, авторегрессионная модель f_{AR}^y соответствует случаю пустой предыстории временных рядов входных сигналов (случаю $h_x = 0$), а регрессионная модель f_R^{xy} соответствует случаю пустого предыстории временных рядов целевых сигналов (случаю $h_y = 0$).

На Рис. 2.1 схематично продемонстрированы принципы построения введенных моделей декодирования временных рядов.

Теорема 1. При определенных условиях (???) - [линейность моделей, уровень корреляции сигнала, обусловленность матрицы] предсказательная модель f^{xy} :

$\mathbb{R}^{h_x \times m} \times \mathbb{R}^{h_y \times r} \rightarrow \mathbb{R}^{p \times r}$ лучше (???) - [меньше ошибки на трейн/вал, устойчивее решение, меньше размерность (простота)], чем авторегрессионная модель f_{AR}^y и регрессионная модель f_{R}^{xy} .

Для постановки задачи декодирования введём предположения о структурах пространств \mathbb{X} и \mathbb{Y} .

Предположение 1. Рассмотрим случай, когда пространства \mathbb{X} и \mathbb{Y} имеют избыточную размерность. Это означает, что объекты \mathbf{x} и \mathbf{y} принадлежат некоторым многообразиям низкой размерности. В простейшем случае такие многообразия могут являться вложениями, линейными подпространствами.

подумать, жирные ли буквы

Определение 12. Назовём пространство $\mathbb{T} \subset \mathbb{R}^l$ скрытым пространством для пространства $\mathbb{X} \in \mathbb{R}^n$ ($l \leq n$), если существуют функция $\varphi_e : \mathbb{X} \rightarrow \mathbb{T}$ и функция $\varphi_d : \mathbb{T} \rightarrow \mathbb{X}$ такие что

$$\mathbf{x} \in \mathbb{X} \quad \exists \mathbf{t} \in \mathbb{T} : \varphi_d(\varphi_e(\mathbf{x})) = \varphi_d(\mathbf{t}) = \mathbf{x}.$$

Функция $\varphi_e(\mathbf{x})$ называется *функцией кодирования* объекта \mathbf{x} , функция $\varphi_d(\mathbf{t})$ называется *функцией декодирования*.

Аналогично введём определение скрытого пространства $\mathbb{U} \subset \mathbb{R}^s$ для целевого пространства \mathbb{Y} , функции кодирования $\psi_e : \mathbb{Y} \rightarrow \mathbb{U}$ и декодирования $\psi_d : \mathbb{U} \rightarrow \mathbb{Y}$

$$\mathbf{y} \in \mathbb{Y} \quad \exists \mathbf{u} \in \mathbb{U} : \psi_d(\psi_e(\mathbf{y})) = \psi_d(\mathbf{u}) = \mathbf{y}.$$

Определение 13. Пространство $\mathbb{T} \subset \mathbb{R}^l$ задаёт *внутреннюю структуру* пространства $\mathbb{X} \in \mathbb{R}^n$, если пространство \mathbb{T} является скрытым для пространства \mathbb{X} .

Определение 14. Между пространствами \mathbb{X} и \mathbb{Y} существует *согласующее отображение*, если существуют пространства \mathbb{T} и \mathbb{U} , задающие внутренние структуры для пространств \mathbb{X} и \mathbb{Y} соответственно, и существует *функция согласования* $g : \mathbb{T} \rightarrow \mathbb{U}$, такая что

$$\mathbf{u} \in \mathbb{U} \quad \exists \mathbf{t} : \mathbf{u} = g(\mathbf{t}).$$

Предположение 2. Предположим, что в задаче предсказания (1.1) пространства \mathbb{T} и \mathbb{U} задают внутреннюю структуру пространств \mathbb{X} и \mathbb{Y} . Предположим также, что для данных скрытых пространств \mathbb{T} и \mathbb{U} существует функция согласования $g : \mathbb{T} \rightarrow \mathbb{U}$. Тогда выполнено

$$\forall \mathbf{y} \in \mathbb{Y} \quad \exists \mathbf{x} \in \mathbb{X} : \mathbf{y} = \psi_d(\mathbf{u}) = \psi_d(g(\mathbf{t})) = \psi_d(g(\varphi_e(\mathbf{x}))),$$

и общая схема задачи декодирования принимает вид следующей коммутативной диаграммы:

$$\begin{array}{ccc} \mathbb{X} \subset \mathbb{R}^n & \xrightarrow{f} & \mathbb{Y} \subset \mathbb{R}^r \\ \varphi_e \swarrow \varphi_d & & \downarrow \psi_d \\ \mathbb{T} \subset \mathbb{R}^l & \xrightarrow{h} & \mathbb{U} \subset \mathbb{R}^s \\ \psi_d \searrow \psi_e & & \end{array} \quad (2.1)$$

$$\begin{array}{ccc} \mathbf{X} \subset \mathbb{R}^n & \xrightarrow{\mathbf{f}} & \mathbf{Y} \subset \mathbb{R}^r \\ & \searrow \mathbf{P} & \swarrow \mathbf{Q} \\ & \mathbf{T}, \mathbf{U} \subset \mathbb{R}^l & \end{array} \quad (2.2)$$

Определение 15. Согласно диаграмме (2.1), определим модель декодирования $f : \mathbb{X} \rightarrow \mathbb{Y}$ как суперпозицию

$$f = \psi_d \circ g \circ \varphi_e. \quad (2.3)$$

2.2. Доказательство корректности работы алгоритмов PLS и CCA

Псевдокод метода регрессии PLS приведен в Алгоритме 1. Алгоритм итеративно на каждом из l шагов вычисляет по одному столбцу \mathbf{t}_k , \mathbf{u}_k , \mathbf{p}_k , \mathbf{q}_k матриц \mathbf{T} , \mathbf{U} , \mathbf{P} , \mathbf{Q} соответственно. После вычисления следующего набора векторов из матриц \mathbf{X} , \mathbf{Y} вычтываются очередные одноранговые аппроксимации. При этом предполагается, что исходные матрицы \mathbf{X} и \mathbf{Y} нормированы (имеют нулевое среднее и единичное среднее отклонение).

Algorithm 1 Алгоритм PLS

Вход: $\mathbf{X}, \mathbf{Y}, l$;

Выход: $\mathbf{T}, \mathbf{P}, \mathbf{Q}$;

- 1: нормировать матрицы \mathbf{X} и \mathbf{Y} по столбцам
 - 2: инициализировать \mathbf{u}_0 (первый столбец матрицы \mathbf{Y})
 - 3: $\mathbf{X}_1 = \mathbf{X}; \mathbf{Y}_1 = \mathbf{Y}$
 - 4: **для** $k = 1, \dots, l$
 - 5: **повторять**
 - 6: $\mathbf{w}_k := \mathbf{X}_k^\top \mathbf{u}_{k-1} / (\mathbf{u}_{k-1}^\top \mathbf{u}_{k-1}); \quad \mathbf{w}_k := \frac{\mathbf{w}_k}{\|\mathbf{w}_k\|}$
 - 7: $\mathbf{t}_k := \mathbf{X}_k \mathbf{w}_k$
 - 8: $\mathbf{c}_k := \mathbf{Y}_k^\top \mathbf{t}_k / (\mathbf{t}_k^\top \mathbf{t}_k); \quad \mathbf{c}_k := \frac{\mathbf{c}_k}{\|\mathbf{c}_k\|}$
 - 9: $\mathbf{u}_k := \mathbf{Y}_k \mathbf{c}_k$
 - 10: **пока** \mathbf{t}_k не стабилизируется
 - 11: $\mathbf{p}_k := \mathbf{X}_k^\top \mathbf{t}_k / (\mathbf{t}_k^\top \mathbf{t}_k), \mathbf{q}_k := \mathbf{Y}_k^\top \mathbf{t}_k / (\mathbf{t}_k^\top \mathbf{t}_k)$
 - 12: $\mathbf{X}_{k+1} := \mathbf{X}_k - \mathbf{t}_k \mathbf{p}_k^\top$
 - 13: $\mathbf{Y}_{k+1} := \mathbf{Y}_k - \mathbf{t}_k \mathbf{q}_k^\top$
-

Вектора \mathbf{t}_k и \mathbf{u}_k из внутреннего цикла Алгоритма 1 содержат информацию о матрице объектов \mathbf{X} и матрице ответов \mathbf{Y} соответственно. Блоки из шагов (6)–(7) и шагов (8)–(9) — аналоги алгоритма PCA для матриц \mathbf{X} и \mathbf{Y} [82]. Последовательное выполнение блоков позволяет учесть взаимную связь между матрицами \mathbf{X} и \mathbf{Y} .

Теоретическое обоснование алгоритма PLS следует из следующих утверждений.

Утверждение 1. Максимизация ковариации между векторами \mathbf{t}_k и \mathbf{u}_k сохраняет дисперсию столбцов матриц \mathbf{X} и \mathbf{Y} и учитывает их линейную зависимость.

Доказательство. Утверждение следует из равенства

$$\text{cov}(\mathbf{t}_k, \mathbf{u}_k) = \text{corr}(\mathbf{t}_k, \mathbf{u}_k) \cdot \sqrt{\text{var}(\mathbf{t}_k)} \cdot \sqrt{\text{var}(\mathbf{u}_k)}.$$

Максимизация дисперсий векторов \mathbf{t}_k и \mathbf{u}_k отвечает за сохранение информации об исходных матрицах, корреляция между векторами отвечает взаимосвязи между \mathbf{X} и \mathbf{Y} . \square

Во внутреннем цикле Алгоритма 1 вычисляются нормированные вектора весов \mathbf{w}_k и \mathbf{c}_k . Из данных векторов строятся матрицы весов \mathbf{W} и \mathbf{C} соответственно.

Утверждение 2. В результате выполнения внутреннего цикла вектора \mathbf{w}_k и \mathbf{c}_k будут собственными векторами матриц $\mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k$ и $\mathbf{Y}_k^\top \mathbf{X}_k \mathbf{X}_k^\top \mathbf{Y}_k$, соответствующими максимальным собственным значениям.

$$\begin{aligned}\mathbf{w}_k &\propto \mathbf{X}_k^\top \mathbf{u}_{k-1} \propto \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{c}_{k-1} \propto \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{t}_{k-1} \propto \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{w}_{k-1}, \\ \mathbf{c}_k &\propto \mathbf{Y}_k^\top \mathbf{t}_k \propto \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{w}_k \propto \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{X}_k^\top \mathbf{u}_{k-1} \propto \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{c}_{k-1},\end{aligned}$$

где символ \propto означает равенство с точностью до мультипликативной константы.

Доказательство. Утверждение следует из того факта, что правила обновления векторов \mathbf{w}_k , \mathbf{c}_k совпадают с итерацией алгоритма поиска максимального собственного значения. Данный алгоритм основан на следующем факте. Если матрица \mathbf{A} диагонализуема, \mathbf{x} — некоторый вектор, то

$$\lim_{k \rightarrow \infty} \mathbf{A}^k \mathbf{x} = \lambda_{\max}(\mathbf{A}) \cdot \mathbf{v}_{\max},$$

где $\lambda_{\max}(\mathbf{A})$ — максимальное собственное значение матрицы \mathbf{A} , \mathbf{v}_{\max} — собственный вектор матрицы \mathbf{A} , соответствующий $\lambda_{\max}(\mathbf{A})$. \square

Утверждение 3. Обновление векторов по шагам (6)–(9) Алгоритма 1 соответствует максимизации ковариации между векторами \mathbf{t}_k и \mathbf{u}_k .

Доказательство. Максимальная ковариация между векторами \mathbf{t}_k и \mathbf{u}_k равна максимальному собственному значению матрицы $\mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k$:

$$\begin{aligned} \max_{\mathbf{t}_k, \mathbf{u}_k} \text{cov}(\mathbf{t}_k, \mathbf{u}_k)^2 &= \max_{\substack{\|\mathbf{w}_k\|=1 \\ \|\mathbf{c}_k\|=1}} \text{cov}(\mathbf{X}_k \mathbf{w}_k, \mathbf{Y}_k \mathbf{c}_k)^2 = \max_{\substack{\|\mathbf{w}_k\|=1 \\ \|\mathbf{c}_k\|=1}} \text{cov}\left(\mathbf{c}_k^\top \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{w}_k\right)^2 = \\ &= \max_{\|\mathbf{w}_k\|=1} \text{cov}\left\|\mathbf{Y}_k^\top \mathbf{X}_k \mathbf{w}_k\right\|^2 = \max_{\|\mathbf{w}_k\|=1} \mathbf{w}_k^\top \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{w}_k = \\ &= \lambda_{\max}\left(\mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k\right), \end{aligned}$$

где $\lambda_{\max}(\mathbf{A})$ — максимальное собственное значение матрицы \mathbf{A} . Применяя утверждение 2, получаем требуемое. \square

После завершения внутреннего цикла на шаге (11) вычисляются вектора \mathbf{p}_k , \mathbf{q}_k проецированием столбцов матриц \mathbf{X}_k и \mathbf{Y}_k на вектор \mathbf{t}_k . Для перехода на следующий шаг необходимо вычесть из матриц \mathbf{X}_k и \mathbf{Y}_k одноранговые аппроксимации $\mathbf{t}_k \mathbf{p}_k^\top$ и $\mathbf{t}_k \mathbf{q}_k^\top$

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{t}_k \mathbf{p}_k^\top = \mathbf{X} - \sum_k \mathbf{t}_k \mathbf{p}_k^\top,$$

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k - \mathbf{t}_k \mathbf{q}_k^\top = \mathbf{Y} - \sum_k \mathbf{t}_k \mathbf{q}_k^\top.$$

При этом каждый следующий вектор \mathbf{t}_k оказывается ортогонален всем векторам \mathbf{t}_i , $i = 1, \dots, k$.

Для получения прогнозов модели и нахождения параметров модели домножим справа формулу (1.8) на матрицу \mathbf{W} . Строки матрицы невязок \mathbf{E} ортогональны столбцам матрицы \mathbf{W} , поэтому

$$\mathbf{XW} = \mathbf{TP}^\top \mathbf{W}.$$

Линейное преобразование между объектами в исходном и латентном пространстве имеет вид

$$\mathbf{T} = \mathbf{XW}^*, \tag{2.4}$$

где $\mathbf{W}^* = \mathbf{W}(\mathbf{P}^\top \mathbf{W})^{-1}$.

Матрица параметров модели 1.4 находится из уравнений (1.9), (2.4)

$$\mathbf{Y} = \mathbf{T}\mathbf{Q}^T + \mathbf{E} = \mathbf{X}\mathbf{W}^*\mathbf{Q}^T + \mathbf{E} = \mathbf{X}\Theta + \mathbf{E}.$$

Таким образом, параметры модели (1.4) равны

$$\Theta = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}\mathbf{Q}^T. \quad (2.5)$$

Финальная модель (2.2.) является линейной, низкоразмерной в скрытом пространстве. Это снижает избыточность данных и повышает стабильность модели.

2.3. Алгоритмы PCA, PLS, CCA, DeepCCA в терминах задачи декодирования (работа Фаруха)

2.3.1. Линейная регрессия

Определение 16. Параметрическая функция $\varphi_1 : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{n \times p}$, переводящая исходные данные в латентное пространство, называется **функцией кодирования**.

Определение 17. Функция $\varphi_2 : \mathbb{R}^{n \times p} \rightarrow \mathbb{R}^{n \times m}$, переводящая данные из латентного пространства в исходное, называется **функцией восстановления**.

Определение 18. Функция $g : \mathbb{R}^{n \times p} \times \mathbb{R}^{n \times p} \rightarrow \mathbb{R}$, связывающая закономерности в низкоразмерных латентных представлениях, называется **функцией согласования**.

Определение 19. Согласование — алгоритмическая процедура максимизации функции согласования.

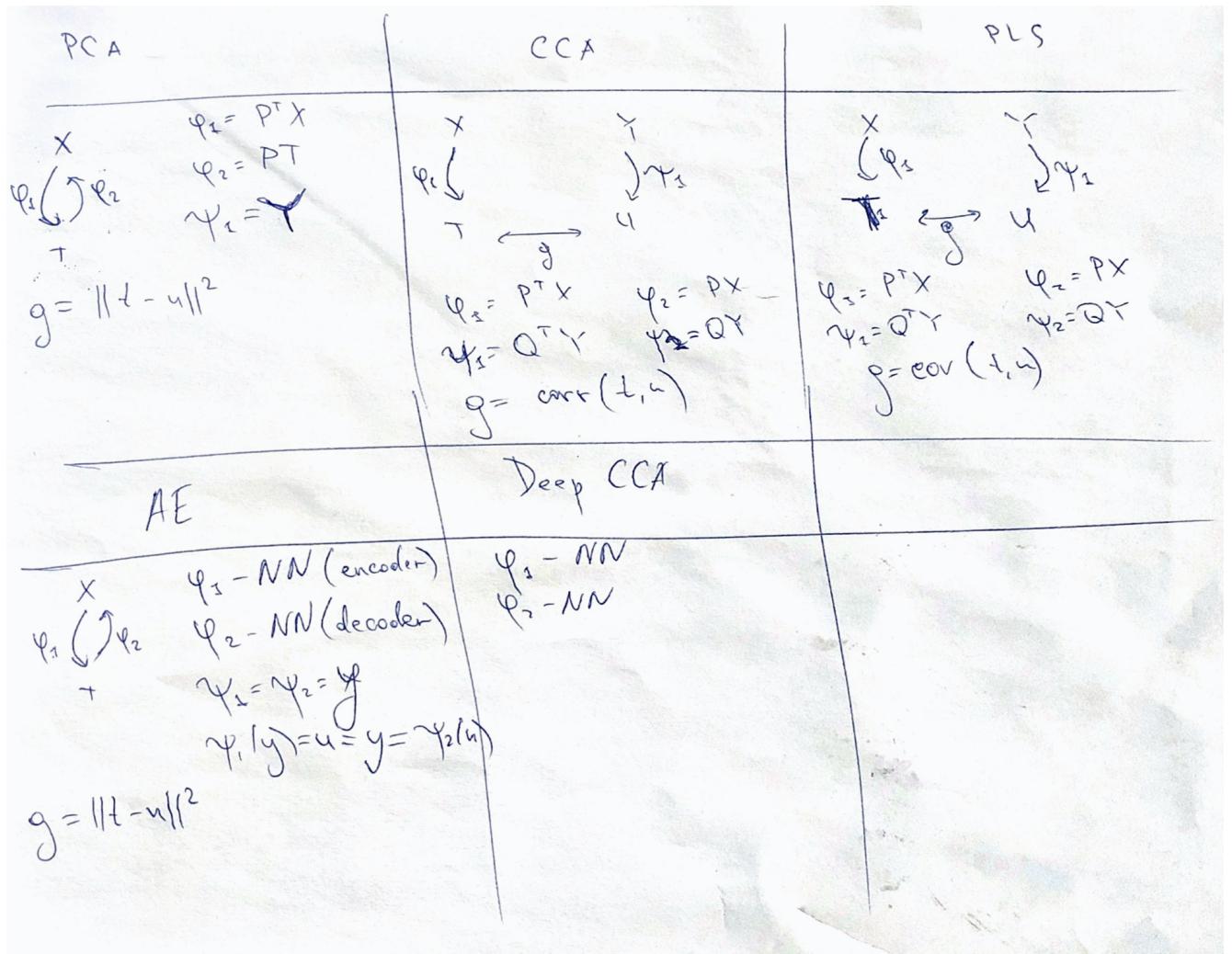


Рис. 2.2: Примеры алгоритмов, работающих по схеме 2.1

2.3.2. Снижение размерности

Коммутативная диаграмма процедуры выбора прогностической модели имеет вид

$$\begin{array}{ccc}
 \mathbf{X}_{n \times m} & \xrightarrow{f} & \mathbf{Y}_{n \times k} \\
 & \varphi_1 \left(\begin{array}{c|c} \varphi_2 & \psi_2 \end{array} \right) \psi_1 & \\
 & \downarrow & \uparrow \\
 \mathbf{T}_{n \times p} & \xleftarrow{g} & \mathbf{U}_{n \times p}
 \end{array} \tag{2.6}$$

где $\varphi_1 : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{n \times p}$ — функция кодирования независимых переменных; $\psi_1 : \mathbb{R}^{n \times k} \rightarrow \mathbb{R}^{n \times p}$ — функция кодирования целевых переменных; $\varphi_2 : \mathbb{R}^{n \times p} \rightarrow$

$\mathbb{R}^{n \times m}$ — функция восстановления независимых переменных; $\psi_2 : \mathbb{R}^{n \times p} \rightarrow \mathbb{R}^{n \times k}$ — функция восстановления целевых переменных; $g : \mathbb{R}^{n \times p} \times \mathbb{R}^{n \times p} \rightarrow \mathbb{R}$ — функция согласования. Матрицы

$$\mathbf{T} = \varphi_1(\mathbf{X}) \in \mathbb{R}^{n \times p}; \quad \mathbf{U} = \psi_1(\mathbf{Y}) \in \mathbb{R}^{n \times p}$$

являются матрицами представлений данных в латентном пространстве низкой размерности.

Оптимальные параметры $\theta_{\varphi_1}^*, \theta_{\psi_1}^*$ для функций кодирования φ_1 и ψ_1 находятся из следующей задачи параметрической оптимизации:

$$(\theta_{\varphi_1}^*, \theta_{\psi_1}^*) = \arg \max_{(\theta_{\varphi_1}, \theta_{\psi_1})} g(\varphi_1(\mathbf{X}; \theta_{\varphi_1}), \psi_1(\mathbf{Y}; \theta_{\psi_1})). \quad (2.7)$$

Так как параметры функции кодирования подбираются из условия максимизации функции согласования (2.7), то после перехода в латентное пространство между \mathbf{T} и \mathbf{U} существует зависимость

$$\mathbf{U} = h(\mathbf{T}) + \boldsymbol{\eta}, \quad (2.8)$$

где $h : \mathbb{R}^{n \times p} \rightarrow \mathbb{R}^{n \times p}$ — функция регрессионной зависимости, $\boldsymbol{\eta}$ — матрица регрессионных ошибок. Оптимальная функция h выбирается минимизацией функции ошибки. Используется квадратичная функция потерь \mathcal{L} на \mathbf{T} и \mathbf{U} :

$$\mathcal{L}(h|\mathbf{T}, \mathbf{U}) = \left\| \mathbf{U}_{n \times p} - h(\mathbf{T}_{m \times p}) \right\|_2^2 \rightarrow \min_h. \quad (2.9)$$

Финальная прогностическая модель имеет вид $\hat{\mathbf{y}} = \psi_2(h(\varphi_1(\mathbf{x})))$, то есть

$$f = \psi_2 \circ h \circ \varphi_1. \quad (2.10)$$

2.3.3. Метод главных компонент

Метод главных компонент (PCA) снижает размерность данных и сохраняет максимальную дисперсию. Линейная модель PCA представляет собой ортого-

нальное линейное преобразование исходного признакового пространства в новое пространство меньшей размерности. Первый базисный вектор строится так, чтобы выборочная дисперсия столбцов проекций матрицы \mathbf{X} была максимальной:

$$\mathbf{p} = \arg \max_{\|\mathbf{p}\|_2=1} [\text{var}(\mathbf{X}\mathbf{p})], \quad (2.11)$$

где $\text{var}(\mathbf{X}\mathbf{p}) = \frac{1}{n}(\mathbf{X}\mathbf{p})^\top \mathbf{X}\mathbf{p}$ обозначает выборочную дисперсию. Последующие базисные векторы находятся итеративно после вычитания проекции на все найденные ранее.

Функция кодирования $\varphi_1 : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{n \times p}$ имеет вид

$$\varphi_1(\mathbf{X}) = \underset{n \times m}{\mathbf{X}} \cdot \underset{m \times p}{\mathbf{P}}^\top, \quad (2.12)$$

где $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_p]$. Метод PCA не согласует независимые переменные и целевые переменные. Из-за этого зависимости в обоих пространствах не учитываются.

2.3.4. Метод частичных наименьших квадратов

Метод частичных наименьших квадратов восстанавливает связь между двумя наборами данных \mathbf{X} и \mathbf{Y} . Алгоритм проецирует \mathbf{X} и \mathbf{Y} на латентное пространство \mathbb{R}^p меньшей размерности. PLS находит матрицы исходных данных \mathbf{X} и \mathbf{Y} в латентном пространстве \mathbf{T} и \mathbf{U} соответственно. Матрица объектов \mathbf{X} и целевая матрица \mathbf{Y} проецируются на латентное пространство следующим образом:

$$\underset{n \times m}{\mathbf{X}} = \underset{n \times p}{\mathbf{T}} \cdot \underset{p \times m}{\mathbf{P}}^\top + \underset{n \times m}{\mathbf{F}}, \quad (2.13)$$

$$\underset{n \times k}{\mathbf{Y}} = \underset{n \times p}{\mathbf{U}} \cdot \underset{p \times k}{\mathbf{Q}}^\top + \underset{n \times k}{\mathbf{E}}, \quad (2.14)$$

где \mathbf{T} и \mathbf{U} — матрицы описания объектов и исходов в латентном пространстве; \mathbf{P} и \mathbf{Q} — матрицы перехода из латентного пространства в исходное; \mathbf{F} , \mathbf{E} — матрицы остатков.

В методе PLS функции кодирования имеют вид

$$\varphi_1(\mathbf{X}) = \mathbf{XW}_x, \quad \psi_1(\mathbf{Y}) = \mathbf{YW}_y,$$

где матрицы весов $\mathbf{W}_x \in \mathbb{R}^{m \times p}$, $\mathbf{W}_y \in \mathbb{R}^{k \times p}$ находятся путем максимизации функции согласования $g(\mathbf{XW}_x, \mathbf{YW}_y) = \text{Cov}(\mathbf{XW}_x, \mathbf{YW}_y)^2$:

$$(\mathbf{W}_x, \mathbf{W}_y) = \arg \max_{\mathbf{W}_y, \mathbf{W}_x} [\text{Cov}(\mathbf{XW}_x, \mathbf{YW}_y)^2], \quad (2.15)$$

где $\text{Cov}(\mathbf{XW}_x, \mathbf{YW}_y)$ — выборочная ковариация.

Функции восстановления принимают вид

$$\varphi_2(\mathbf{T}) = \mathbf{TP}^\top, \quad \psi_2(\mathbf{U}) = \mathbf{UQ}^\top.$$

2.3.5. Канонический анализ корреляций

Канонический анализ корреляций находит два набора базисных векторов $\{\mathbf{w}_{x_i}\}_{i=1}^p$, $\mathbf{w}_x \in \mathbb{R}^m$, и $\{\mathbf{w}_{y_i}\}_{i=1}^p$, $\mathbf{w}_y \in \mathbb{R}^k$, один для матрицы \mathbf{X} , другой для матрицы \mathbf{Y} , так чтобы коэффициент корреляции между проекциями переменных на эти базисные векторы был максимальным. Функция согласования для СКА имеет вид

$$g(\mathbf{XW}_x, \mathbf{YW}_y) = \text{corr}(\mathbf{XW}_x, \mathbf{YW}_y),$$

где $\text{corr}(\mathbf{Xw}_x, \mathbf{Yw}_y)$ — коэффициент корреляции между векторами.

Таким образом, функции кодирования имеют вид

$$\varphi_1(\mathbf{X}) = \mathbf{XW}_x, \quad \psi_1(\mathbf{Y}) = \mathbf{YW}_y,$$

где первые столбцы матриц весов находятся как векторы, максимизирующие функцию согласования g . Далее ищутся векторы, максимизирующие g , но с ограничением, что они не коррелируют с первой парой векторов. Процедура продолжается до тех пор, пока число векторов не станет равным p .

2.3.6. Нелинейный канонический анализ корреляций

Нелинейный канонический анализ корреляций — нелинейная модификация ССА. Метод Deep ССА преобразует исходные данные с помощью нейронной сети таким образом, что результирующее представление становится согласованным. В данной работе рассматриваются следующие нелинейные функции кодирования и восстановления:

$$\begin{aligned}\mathbf{T} &= \varphi_1(\mathbf{X}) = \mathbf{W}_{\mathbf{x}}^L \sigma(\dots \sigma(\mathbf{W}_{\mathbf{x}}^2 \sigma(\mathbf{X} \mathbf{W}_{\mathbf{x}}^1)) \dots), \\ \mathbf{U} &= \psi_1(\mathbf{Y}) = \mathbf{W}_{\mathbf{y}}^L \sigma(\dots \sigma(\mathbf{W}_{\mathbf{y}}^2 \sigma(\mathbf{Y} \mathbf{W}_{\mathbf{y}}^1)) \dots), \\ \mathbf{X} &= \varphi_2(\mathbf{X}) = \mathbf{W}_{\mathbf{t}}^L \sigma(\dots \sigma(\mathbf{W}_{\mathbf{t}}^2 \sigma(\mathbf{T} \mathbf{W}_{\mathbf{t}}^1)) \dots), \\ \mathbf{Y} &= \psi_2(\mathbf{Y}) = \mathbf{W}_{\mathbf{u}}^L \sigma(\dots \sigma(\mathbf{W}_{\mathbf{u}}^2 \sigma(\mathbf{U} \mathbf{W}_{\mathbf{u}}^1)) \dots).\end{aligned}$$

Каждая функция представляет нейронную сеть с L скрытыми слоями.

Требуется найти такие параметры, при которых функция согласования g достигает своего максимума:

$$g(\mathbf{T}, \mathbf{U}) \rightarrow \max_{\mathbf{W}}, \quad (2.16)$$

где $\mathbf{W} = \{\{\mathbf{W}_{\mathbf{x}}^i\}_{i=1}^L, \{\mathbf{W}_{\mathbf{y}}^i\}_{i=1}^L, \{\mathbf{W}_{\mathbf{t}}^i\}_{i=1}^L, \{\mathbf{W}_{\mathbf{u}}^i\}_{i=1}^L\}$.

Рассмотрим случай квадратичной функции потерь:

$$\mathcal{L}(f, \mathbf{X}, \mathbf{Y}) = \sum_{i=1}^m \|\mathbf{y}_i - f(\mathbf{x}_i)\|^2.$$

Утверждение 4. Пусть функция декодирования ψ_d модели декодирования f (2.3) Липшицева с константой L . Тогда

$$\mathcal{L}(f, \mathbf{X}, \mathbf{Y}) \leq L \cdot \mathcal{L}(g, \mathbf{T}, \mathbf{U}).$$

Доказательство.

$$\begin{aligned}\mathcal{L}(f, \mathbf{X}, \mathbf{Y}) &= \sum_{i=1}^m \|\psi_d(g(\varphi_e(\mathbf{x}_i))) - \mathbf{y}_i\|^2 = \sum_{i=1}^m \|\psi_d(g(\mathbf{t}_i)) - \psi_d(\mathbf{u}_i)\|^2 \\ &\leq L \sum_{i=1}^m \|g(\mathbf{t}_i) - \mathbf{u}_i\|^2 = L \cdot \mathcal{L}(g, \mathbf{T}, \mathbf{U}).\end{aligned}$$

□

Теорема 2. Рассмотрим две модели:

1. Первая модель f_1 доставляет минимум ошибке $\mathcal{L}(f, \mathbf{X}, \mathbf{Y})$

$$f_1 = \arg \min_f \mathcal{L}(f, \mathbf{X}, \mathbf{Y}).$$

При этом $\mathbf{Y} = f_1(\mathbf{X}) + \mathbf{E}_1$, где $\mathbf{E}_1 \in \mathbb{R}^{m \times r}$ — матрица ошибок модели f_1 .

2. Вторая модель f_2 — модель декодирования (2.3) с Липшицевой функцией декодирования ψ_d с константой L и функцией согласования, удовлетворяющей условию

$$g = \arg \min_g \mathcal{L}(g, \mathbf{T}, \mathbf{U}).$$

При этом $\mathbf{U} = g(\mathbf{T}) + \mathbf{E}_2$, где $\mathbf{E}_2 \in \mathbb{R}^{m \times s}$ — матрица ошибок функции согласования g .

Пусть для константы Липшица L функции декодирования ψ_d выполнено следующее условие

$$L \leq \frac{\|\mathbf{E}_1\|_F^2}{\|\mathbf{E}_2\|_F^2}. \quad (2.17)$$

Тогда функция ошибки $\mathcal{L}(f_2, \mathbf{X}, \mathbf{Y})$ модели f_2 не превосходит функции ошибки $\mathcal{L}(f_1, \mathbf{X}, \mathbf{Y})$ модели f_1 .

Доказательство.

$$\begin{aligned}\mathcal{L}(f_1, \mathbf{X}, \mathbf{Y}) &= \sum_{i=1}^m \|\mathbf{f}_1(\mathbf{x}_i) - \mathbf{y}_i\|^2 = \|\mathbf{E}_1\|_F^2. \\ \mathcal{L}(g, \mathbf{T}, \mathbf{U}) &= \sum_{i=1}^m \|\mathbf{g}(\mathbf{t}_i) - \mathbf{u}_i\|^2 = \|\mathbf{E}_2\|_F^2.\end{aligned}$$

Воспользуемся утверждением 4 и условием (2.17)

$$\mathcal{L}(f_2, \mathbf{X}, \mathbf{Y}) \leq L \cdot \mathcal{L}(g, \mathbf{T}, \mathbf{U}) = L \|\mathbf{E}_2\|_F^2 \leq \|\mathbf{E}_1\|_F^2 = \mathcal{L}(f_1, \mathbf{X}, \mathbf{Y}).$$

□

2.4. Вычислительный эксперимент

Временные ряды электроэнергии состоят из почасовых записей (52512 наблюдений). Стока матрицы \mathbf{X} — локальная история сигнала за одну неделю $n = 24 \times 7$. Стока матрицы \mathbf{Y} — локальный прогноз потребления электроэнергии в следующие 24 часа $r = 24$. В этом случае матрицы \mathbf{X} и \mathbf{Y} являются авторегрессионными матрицами.

Вычислительный эксперимент также проводился на данных электрокортикограмм (ECoG) из проекта NeuroTycho [83]. Данные ECoG состоят из 32-канальных сигналов напряжения, снятых с головного мозга. Цель состоит в предсказании по входному сигналу ECoG 3D позиции рук в следующие моменты времени. Исходные сигналы напряжения преобразуются в пространственно-временное представление с помощью вейвлет-преобразования с материнским вейвлетом Морле. Процедура извлечения признаков из исходных данных подробно описана в [84, 70]. Описание исходного сигнала в каждый момент времени имеет размерность 32 (каналы) \times 27 (частоты) = 864. Каждый объект представляет собой локальный отрезок времени длительностью $\Delta t = 1s$. Временной шаг между объектами $\delta t = 0.05s$. Матрицы имеют размеры $\mathbf{X} \in \mathbb{R}^{18900 \times 864}$ и $\mathbf{Y} \in \mathbb{R}^{18900 \times 3k}$, где k - число отсчётов времени прогнозирования. Данные разбиты на тренировочную и тестовую части в соотношении 0,67. Пример исходных сигналов мозга и соответствующей траектории руки показан на рисунке ??.

Введём среднеквадратичную ошибку для некоторых матриц $\mathbf{A} = [a_{ij}]$ и

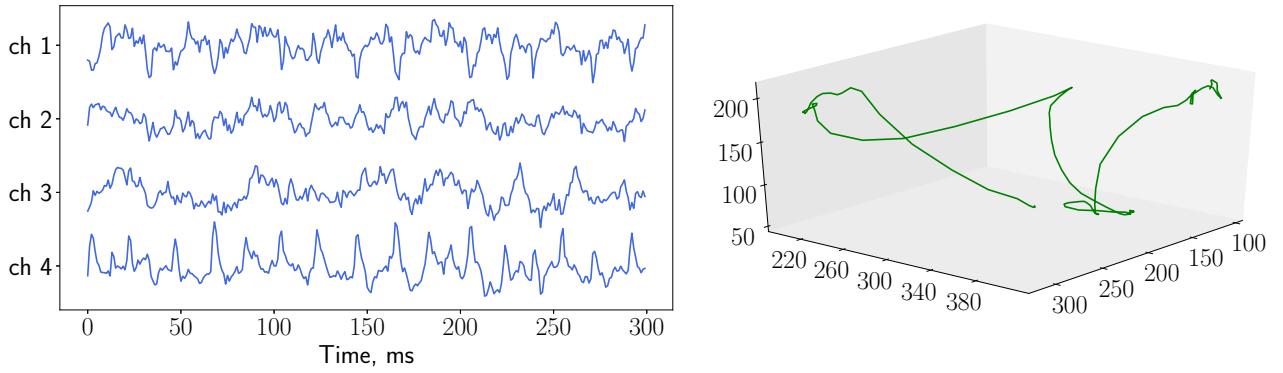


Рис. 2.3: Сигналы мозга (левый график) и 3D координаты руки (правый график)

$$\mathbf{B} = [b_{ij}]$$

$$\text{MSE}(\mathbf{A}, \mathbf{B}) = \sum_{i,j} (a_{ij} - b_{ij})^2.$$

Для оценивания качества аппроксимации вычисляется значение нормированной среднеквадратичной ошибки

$$\text{NMSE}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{\text{MSE}(\mathbf{Y}, \hat{\mathbf{Y}})}{\text{MSE}(\mathbf{Y}, \bar{\mathbf{Y}})}, \quad (2.18)$$

где $\hat{\mathbf{Y}}$ — прогноз модели, $\bar{\mathbf{Y}}$ — константный прогноз средним значением по столбцам матрицы.

Данные потребления электроэнергии

Для нахождения оптимальной размерности l латентного пространства все данные потребления электроэнергии были разбиты на обучающую и валидационную части. Обучающая выборка состоит из 700 объектов, валидационная из 370. Зависимость нормированной квадратичной ошибки (2.18) от размерности l латентного пространства представлена на Рис. 2.4. Сначала ошибка резко падает при увеличении размерности скрытого пространства, а затем стабилизируется.

Минимальная ошибка наблюдается при $l = 14$. Построим прогноз потребления электроэнергии при данном l . Результат аппроксимации изображен на

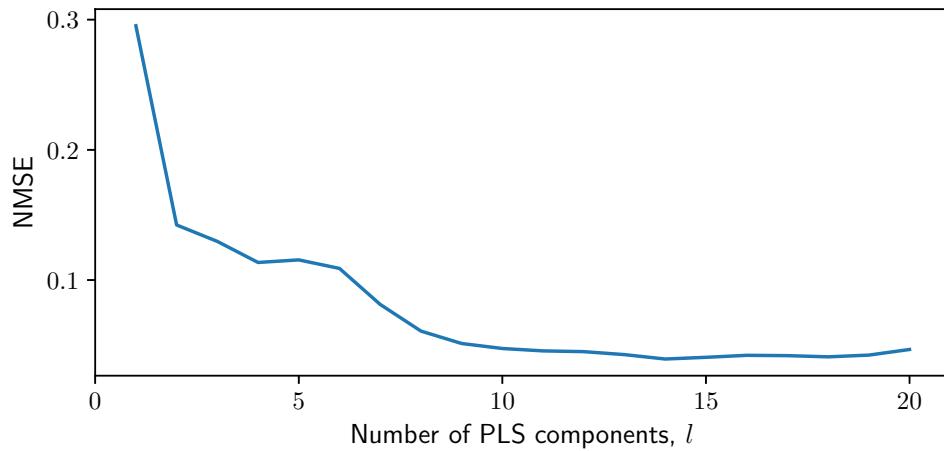


Рис. 2.4: Прогноз потребления электроэнергии алгоритмом PLS при размерности латентного пространства $l=14$

Рис. 2.5. Алгоритм PLS восстановил авторегрессионную зависимость и обнаружил дневную сезонность.

Данные электрокардиограммы

На Рис. 2.6 представлена зависимость нормированной квадратичной ошибки (2.18) от размерности латентного пространства. Ошибка аппроксимации меняется незначительно при $l > 5$. Таким образом совместное описание пространственно-временного спектрального представления объектов и пространственного положения руки может быть представлено вектором размерности $l \ll n$. Зафиксируем $l = 5$. Пример аппроксимации положения руки изображен на Рис. 2.7. Сплошными линиями изображены истинные координаты руки по всем осям, пунктирными линиями показана аппроксимация методом PLS.

2.5. Вычислительный эксперимент из работы Фаруха

Цель вычислительного эксперимента – сравнительный анализ рассматриваемых моделей. Рассматриваются данные, для которых сложность класса ли-

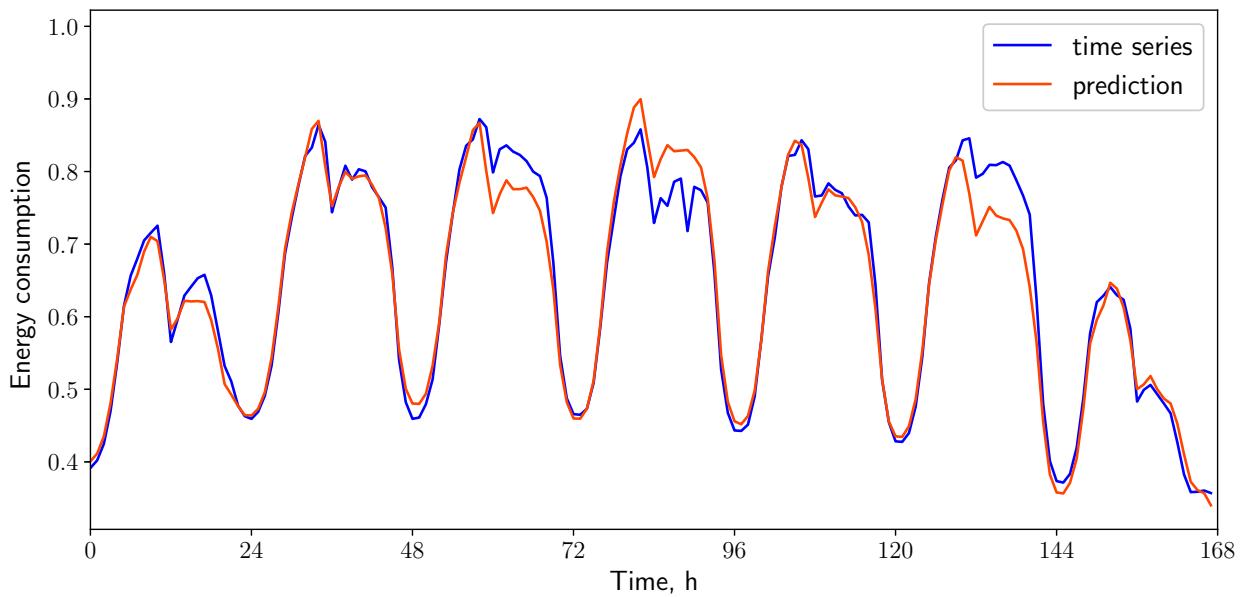


Рис. 2.5: Зависимость ошибки от размерности латентного пространства для данных потребления электроэнергии

нейных методов неадекватно низка. Нелинейные модели позволяют получить точный прогноз при адекватной сложности. В рамках вычислительного эксперимента написан программный комплекс для решения поставленных задач [?].

2.5.1. Анализ нелинейных зависимостей в задаче фильтрации шума

Проведем сравнение качества Deep CCA и CCA на задаче классификации зашумленных цифровых изображений, представленных на рис. 1. Для этого используется набор данных MNIST [?], который состоит из 70 000 цифровых изображений 28×28 образцов рукописного написания цифр. Предлагается получить два новых набора данных \mathbf{X} и \mathbf{Y} следующим образом. Первый набор получается поворотом исходных изображений на угол в диапазоне $[-\frac{\pi}{4}, \frac{\pi}{4}]$. Для получения второго набора данных для каждой картинки из первого набора данных ставится в соответствие случайнym образом картинка с той же цифрой, но с добавлением независимого случайногo шума, распределенного равномерно на отрезке $[0,1]$.

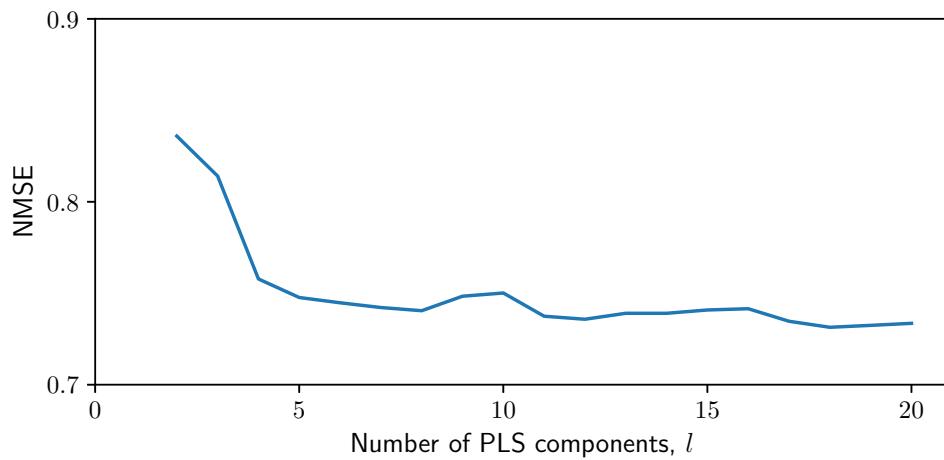


Рис. 2.6: Зависимость ошибки от размерности латентного пространства для данных ECoG

Применив к двум новым наборам данных DeepCCA или CCA, получаем новое низкоразмерное признаковое пространство, которое игнорирует шумы в исходных данных. Таким образом, получаем функции кодирования φ_1 и ψ_1 для исходных наборов данных. На новых признаках, полученных разными моделями (DeepCCA и CCA), для первого набора данных, то есть на данных после применения функции кодирования φ_1 к первому набору исходных данных, обучим линейный SVM-классификатор. Показателем эффективности будет точность классификации линейного SVM на тестовых данных. В случае построения адекватного скрытого пространства полученные образы объектов будут линейно разделимы. Результаты эксперимента приведены в табл. 2.1. Модель Deep CCA представляет собой нейронную сеть с $L = 3$ скрытыми слоями. Точ-

Таблица 2.1: Точность классификации линейного SVM для алгоритмов Deep CCA и CCA

Скользящий контроль	Deep CCA ($L = 3$)	CCA
Валидация	92,74%	76,21%
Тест	92,14%	76,07%

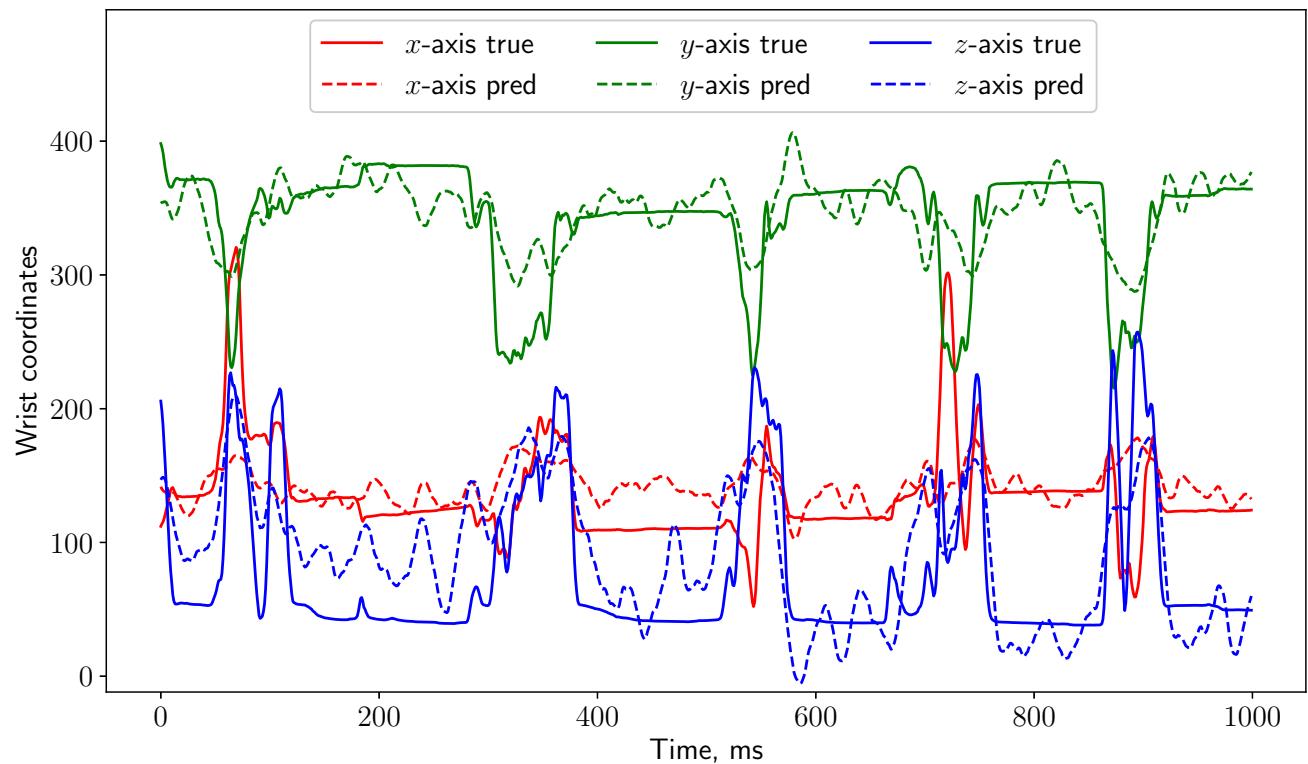


Рис. 2.7: Прогноз движения руки по данным ECoG алгоритмом PLS при раз мерности латентного пространства $l = 5$

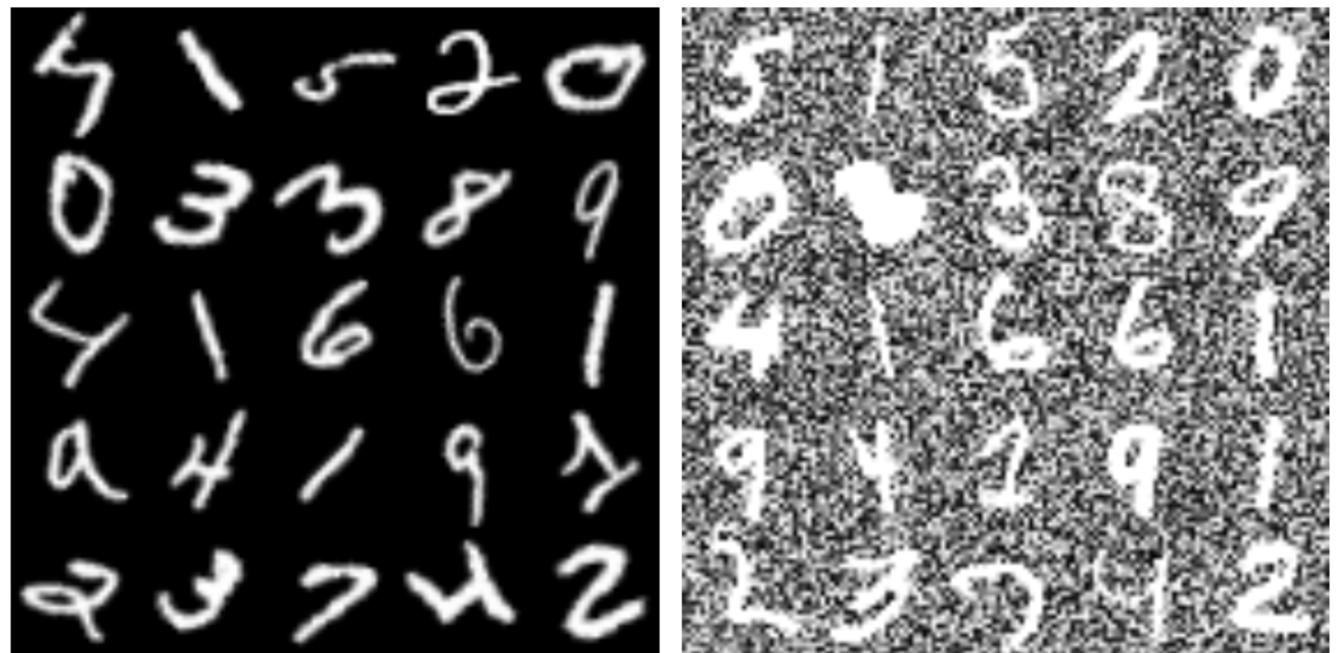


Рис. 2.8: Зашумленные изображения из набора данных MNIST

ность классификации нелинейной модели существенно выше линейного алго-

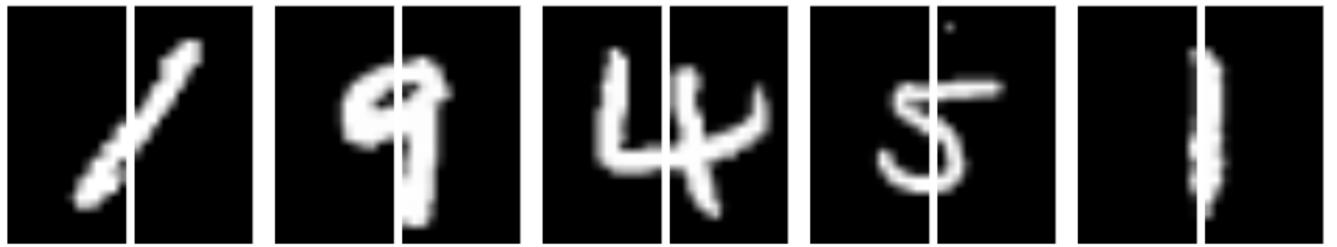


Рис. 2.9: Набор данных MNIST, в котором каждое изображение разделено пополам

ритма ССА.

2.5.2. Анализ нелинейных моделей для восстановления изображений

Для анализа процедуры согласования проведен вычислительный эксперимент с предложенными нелинейными моделями. Для снижения размерности пространства используются нейросетевые модели автокодировщика с согласованием скрытого пространства (2.16). В качестве базовых моделей используются модель автокодировщика без согласования скрытых пространств, а также линейный PLS. В качестве исходного набора данных используется набор данных MNIST [?]. Каждое изображение поделено на левую и правую части, как показано на рис. 2.9. Модель по левому изображению восстанавливает правое изображение.

Модель EncNet1 — нейронная сеть с нелинейными функциями активации, которая обучается на данных после преобразования их автоэнкодером. Модель LinNet1 — нейронная сеть с одним линейным слоем, которая также обучается на преобразованных данных. Для EncNet1 и LinNet1 автоэнкодеры для объектов и ответов используют совместную функцию потерь, которая связывает выходы энкодеров. Модели EncNet2 и LinNet2 устроены аналогично EncNet1 и LinNet1 соответственно, но в автоэнкодерах нет совместной функции потерь. Модель DumbNet — нейронная сеть, которая обучается на исходных данных и имеет

такую же структуру, что и EncNet, то есть имеет такое же число слоев и в каждом слое такое же количество нейронов, что и у EncNet.

Для оценки качества моделей вычислялась среднеквадратичная ошибка. Примеры восстановленных изображений показаны на рис. 2.10. Качество моделей, а также их сложность представлены в табл. 2.2. На рис. 2.10 продемонстрировано, что предложенные модели EncNet и LinNet позволяют получить более четкие и различимые изображения, в отличие от базовой нелинейной модели DumbNet и линейной модели PLS. Несмотря на заметное улучшение визуального качества изображений, ошибка предложенных моделей выше, чем у модели DumbNet. Авторы предполагают, что это связано с тем, что среднеквадратичная ошибка оказалась неадекватной метрикой в пространстве изображений. Нахождение оптимальной метрики для оценки качества предложенных алгоритмов может быть одним из возможных направлений развития текущей работы.

Таблица 2.2: Квадратичная ошибка для нелинейных моделей в задаче восстановления правой части изображения по левой

	EncNet1	LinNet1	EncNet2	LinNet2	DumbNet	PLS
Число параметров, тыс.	283	239	283	239	283	–
Ошибка на тесте	0,147	0,235	0,149	0,236	0,128	0,188

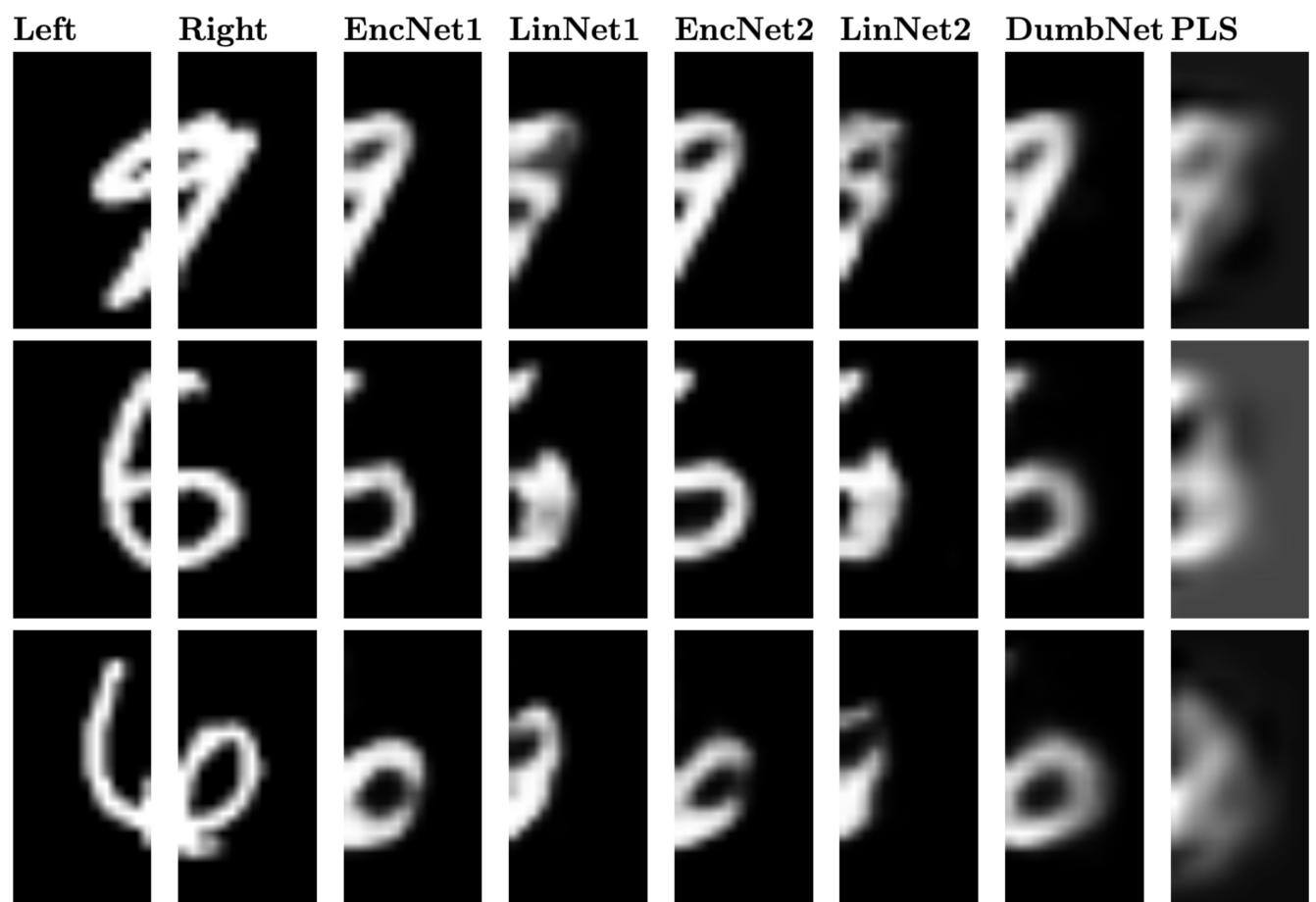


Рис. 2.10: Пример реконструкции правой части изображения по левой для рассматриваемых моделей

Глава 3

Выбор признаков в задаче декодирования сигналов

Задача выбора признаков заключается в поиске оптимального подмножества признаков \mathcal{A} среди всех возможных $2^n - 1$ вариантов. Существует взаимооднозначное отображение между подмножеством \mathcal{A} и булевым вектором $\mathbf{a} \in \{0, 1\}^n$, компоненты которого указывают, выбран ли признак. Для нахождения оптимального вектора \mathbf{a} введем функцию ошибки выбора признаков $S(\mathbf{a}, \mathbf{X}, \mathbf{Y})$. Проблема выбора признаков принимает вид:

$$\mathbf{a} = \arg \min_{\mathbf{a}' \in \{0,1\}^n} S(\mathbf{a}', \mathbf{X}, \mathbf{Y}). \quad (3.1)$$

Целью выбора признаков является построение функции $S(\mathbf{a}, \mathbf{X}, \mathbf{Y})$. Конкретные примеры данной функции для рассматриваемых алгоритмов выбора признаков приведены ниже и обобщены в таблице 3.1.

Задача (3.1) имеет дискретную область определения $\{0, 1\}^n$. Для решения данной задачи применяется релаксация задачи (3.1) к непрерывной области определения $[0, 1]^n$. Релаксированная задача выбора функции имеет следующий вид:

$$\mathbf{z} = \arg \min_{\mathbf{z}' \in [0,1]^n} S(\mathbf{z}', \mathbf{X}, \mathbf{Y}). \quad (3.2)$$

Здесь, компоненты вектора \mathbf{z} – значения нормированных коэффициентов значимости признаков. Сначала решается задача (3.2), для получения вектора значимостей \mathbf{z} . Затем решение (3.1) восстанавливается с помощью отсечения по порогу следующим образом:

$$\mathbf{a} = [a_j]_{j=1}^n, \quad a_j = \begin{cases} 1, & z_j > \tau; \\ 0, & \text{в противном случае.} \end{cases} \quad (3.3)$$

τ – гиперпараметр, который может быть подобран вручную или выбран с помощью кросс-валидации.

Как только решение \mathbf{a} задачи (3.1) получено, задача (1.5) принимает вид:

$$\mathcal{L}(\boldsymbol{\Theta}_{\mathcal{A}}, \mathbf{X}_{\mathcal{A}}, \mathbf{Y}) = \left\| \mathbf{Y} - \mathbf{X}_{\mathcal{A}} \boldsymbol{\Theta}_{\mathcal{A}}^{\top} \right\|_2^2 \rightarrow \min_{\boldsymbol{\Theta}_{\mathcal{A}}},$$

где индекс \mathcal{A} обозначает подматрицу со столбцами, индексы которых содержатся в \mathcal{A} .

3.1. Выбор признаков с помощью квадратичного программирования

Если между столбцами матрицы плана \mathbf{X} существует линейная зависимость, то решение задачи линейной регрессии

$$\|\boldsymbol{\nu} - \mathbf{X}\boldsymbol{\theta}\|_2^2 \rightarrow \min_{\boldsymbol{\theta} \in \mathbb{R}^n}. \quad (3.4)$$

оказывается неустойчивым. Методы выбора признаков находят подмножество $\mathcal{A} \in \{1, \dots, n\}$ оптимальных столбцов \mathbf{X} .

Алгоритм QPFS выбирает некоррелированные признаки, релевантные целевому вектору $\boldsymbol{\nu}$. Чтобы формализовать этот подход, введем две функции: $\text{Sim}(\mathbf{X})$ и $\text{Rel}(\mathbf{X}, \boldsymbol{\nu})$. $\text{Sim}(\mathbf{X})$ контролирует избыточность между признаками, $\text{Rel}(\mathbf{X}, \boldsymbol{\nu})$ содержит релевантности между каждым признаком и целевым вектором. Мы хотим минимизировать функцию Sim и максимизировать Rel одновременно.

QPFS предлагает явный способ построения функций Sim и Rel . Алгоритм минимизирует следующую функцию ошибки

$$\underbrace{\mathbf{z}^{\top} \mathbf{Q} \mathbf{z}}_{\text{Sim}} - \alpha \cdot \underbrace{\mathbf{b}^{\top} \mathbf{z}}_{\text{Rel}} \rightarrow \min_{\substack{\mathbf{z} \in \mathbb{R}_+^n \\ \|\mathbf{z}\|_1=1}}.$$
(3.5)

Элементы матрицы $\mathbf{Q} \in \mathbb{R}^{n \times n}$ содержат коэффициенты попарного сходства между признаками. Вектор $\mathbf{b} \in \mathbb{R}^n$ выражает сходство между каждым признаком и целевым вектором $\boldsymbol{\nu}$. Нормированный вектор \mathbf{z} отражает значимость

каждого признака. Функция ошибки (3.5) штрафует зависимые признаки функцией Sim и штрафует признаки, не релевантные к целевой переменной функцией Rel. Параметр α позволяет контролировать компромисс между Sim и Rel. Авторы оригинальной статьи QPFS [12] предложили способ выбора α , чтобы уравновесить вклад членов Sim(\mathbf{X}) и Rel($\mathbf{X}, \boldsymbol{\nu}$)

$$\alpha = \frac{\bar{\mathbf{Q}}}{\bar{\mathbf{Q}} + \bar{\mathbf{b}}}, \quad \text{где } \bar{\mathbf{Q}} = \text{mean}(\mathbf{Q}), \quad \bar{\mathbf{b}} = \text{mean}(\mathbf{b}).$$

Чтобы выделить оптимальное подмножество признаков, применяется отсечение по порогу (3.3).

Для измерения сходства используется выборочный коэффициент корреляции Пирсона между парами признаков для функции Sim, и между признаками и целевым вектором для функции Rel:

$$\mathbf{Q} = [| \text{corr}(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j) |]_{i,j=1}^n, \quad \mathbf{b} = [| \text{corr}(\boldsymbol{\chi}_i, \boldsymbol{\nu}) |]_{i=1}^n. \quad (3.6)$$

Здесь

$$\text{corr}(\boldsymbol{\chi}, \boldsymbol{\nu}) = \frac{\sum_{i=1}^m (\boldsymbol{\chi}_i - \bar{\boldsymbol{\chi}})(\boldsymbol{\nu}_i - \bar{\boldsymbol{\nu}})}{\sqrt{\sum_{i=1}^m (\boldsymbol{\chi}_i - \bar{\boldsymbol{\chi}})^2} \sqrt{\sum_{i=1}^m (\boldsymbol{\nu}_i - \bar{\boldsymbol{\nu}})^2}}.$$

Другие способы определения \mathbf{Q} и \mathbf{b} рассматриваются в [4]. В работе [4] показано, что алгоритм QPFS превосходит многие существующие алгоритмы выбора функций на различных критериях качества.

Задача (3.5) является выпуклой, если матрица \mathbf{Q} является неотрицательно определенной. В общем случае это не всегда верно. Чтобы удовлетворить этому условию спектр матрицы \mathbf{Q} смещается, и матрица \mathbf{Q} заменяется на $\mathbf{Q} - \lambda_{\min} \mathbf{I}$, где λ_{\min} является минимальным собственным значением \mathbf{Q} .

3.2. Многоиндексный метод выбора признаков

В данном разделе описаны предлагаемые методы выбора признаков для случая нескольких многомерной целевой переменной. В этом случае компоненты

целевой переменной могут коррелировать между собой. Предлагаются алгоритмы, учитывающие зависимости как во входном, так и в целевом пространствах.

3.2.1. Агрегация релевантностей целевых переменных

В работе [1], чтобы применить алгоритм QPFS к многомерному случаю ($r > 1$), релевантности признаков агрегируются по всем r компонентам. Член $\text{Sim}(\mathbf{X})$ остаётся без изменений, матрица \mathbf{Q} определяется как (3.6). Вектор \mathbf{b} агрегируется по всем компонентам целевой переменной и определяется как

$$\mathbf{b} = \left[\sum_{k=1}^r |\text{corr}(\boldsymbol{\chi}_i, \boldsymbol{\nu}_k)| \right]_{i=1}^n.$$

Недостатком такого подхода является отсутствие учёта зависимостей в столбцах матрицы \mathbf{Y} . Рассмотрим следующий пример:

$$\mathbf{X} = [\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, \boldsymbol{\chi}_3], \quad \mathbf{Y} = [\underbrace{\boldsymbol{\nu}_1, \boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_1}_{r-1}, \boldsymbol{\nu}_2].$$

Пусть матрица \mathbf{X} содержит 3 столбца, матрица $\mathbf{Y} - r$ столбцов, где первые $r - 1$ компонент целевой переменной идентичны. Попарные сходства признаков задаются матрицей \mathbf{Q} . Матрица \mathbf{B} содержит попарные сходства признаков и целевых столбцов. Вектор \mathbf{b} получен суммированием матрицы \mathbf{B} по столбцами

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.8 \\ 0 & 0.8 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.4 & \dots & 0.4 & 0 \\ 0.5 & \dots & 0.5 & 0.8 \\ 0.8 & \dots & 0.8 & 0.1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} (r-1) \cdot 0.4 + 0 \\ (r-1) \cdot 0.5 + 0.8 \\ (r-1) \cdot 0.8 + 0.1 \end{bmatrix}. \quad (3.7)$$

Пусть необходимо выбрать только 2 признака. В данном случае оптимальным подмножеством признаков является $[\boldsymbol{\chi}_1, \boldsymbol{\chi}_2]$. Признак $\boldsymbol{\chi}_2$ предсказывает второй целевой столбец $\boldsymbol{\nu}_2$, комбинация признаков $\boldsymbol{\chi}_1, \boldsymbol{\chi}_2$ прогнозирует первый целевой столбец $\boldsymbol{\nu}_1$. Алгоритм QPFS для $r = 2$ дает решение $\mathbf{z} = [0.37, 0.61, 0.02]$.

Это совпадает с описанным решением. Однако, если добавить коллинеарные столбцы в матрицу \mathbf{Y} и увеличить r до 5, то решением QPFS будет $\mathbf{z} = [0.40, 0.17, 0.43]$. Здесь потерян признак χ_2 и выбран избыточный признак χ_3 . В следующих подразделах предлагаются обобщения алгоритма QPFS, которые позволяют бороться с проблемой данного примера.

3.2.2. Симметричный учёт значимости признаков и целевых переменных

Чтобы учесть зависимости в столбцах матрицы \mathbf{Y} , обобщим функцию QPFS (3.5) для многомерного случая ($r > 1$). Добавим член $\text{Sim}(\mathbf{Y})$ и изменим член $\text{Rel}(\mathbf{X}, \mathbf{Y})$ следующим образом:

$$\alpha_1 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x}_{\text{Sim}(\mathbf{X})} - \alpha_2 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y}_{\text{Rel}(\mathbf{X}, \mathbf{Y})} + \alpha_3 \cdot \underbrace{\mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y}_{\text{Sim}(\mathbf{Y})} \rightarrow \min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n, \mathbf{1}_n^\top \mathbf{z}_x = 1 \\ \mathbf{z}_y \geq \mathbf{0}_r, \mathbf{1}_r^\top \mathbf{z}_y = 1}} . \quad (3.8)$$

Определим элементы матриц $\mathbf{Q}_x \in \mathbb{R}^{n \times n}$, $\mathbf{Q}_y \in \mathbb{R}^{r \times r}$ и $\mathbf{B} \in \mathbb{R}^{n \times r}$ следующим образом:

$$\mathbf{Q}_x = [|\text{corr}(\chi_i, \chi_j)|]_{i,j=1}^n, \quad \mathbf{Q}_y = [|\text{corr}(\nu_i, \nu_j)|]_{i,j=1}^r, \quad \mathbf{B} = [|\text{corr}(\chi_i, \nu_j)|]_{i=1,\dots,n, j=1,\dots,r}.$$

Вектор \mathbf{z}_x содержит коэффициенты значимости признаков, \mathbf{z}_y – коэффициенты значимости целевых столбцов. Коррелированные целевые столбцы штрафуются членом $\text{Sim}(\mathbf{Y})$ и получают более низкие значения значимости.

Коэффициенты α_1 , α_2 , и α_3 контролируют влияние каждого члена на функцию (3.8) и удовлетворяют следующим условиям:

$$\alpha_1 + \alpha_2 + \alpha_3 = 1, \quad \alpha_i \geq 0, \quad i = 1, 2, 3.$$

Утверждение 5. Баланс между $\text{Sim}(\mathbf{X})$, $\text{Rel}(\mathbf{X}, \mathbf{Y})$ и $\text{Sim}(\mathbf{Y})$ в задаче (3.8) достигается при:

$$\alpha_1 \propto \overline{\mathbf{Q}}_y \overline{\mathbf{B}}; \quad \alpha_2 \propto \overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y; \quad \alpha_3 \propto \overline{\mathbf{Q}}_x \overline{\mathbf{B}}. \quad (3.9)$$

Доказательство. Значения α_1 , α_2 , и α_3 получаются путем решения следующих уравнений:

$$\begin{aligned}\alpha_1 + \alpha_2 + \alpha_3 &= 1; \\ \alpha_1 \bar{\mathbf{Q}}_x &= \alpha_2 \bar{\mathbf{B}} = \alpha_3 \bar{\mathbf{Q}}_y.\end{aligned}$$

Здесь $\bar{\mathbf{Q}}_x$, $\bar{\mathbf{B}}$ и $\bar{\mathbf{Q}}_y$ соответствующих матриц \mathbf{Q}_x , \mathbf{B} и \mathbf{Q}_y - средние значения членов $\text{Sim}(\mathbf{X})$, $\text{Rel}(\mathbf{X}, \mathbf{Y})$ и $\text{Sim}(\mathbf{Y})$. \square

Для изучения зависимости $\text{Sim}(\mathbf{Y})$ на функцию (3.8), зафиксируем соотношение между α_1 и α_2 :

$$\alpha_1 = \frac{(1 - \alpha_3)\bar{\mathbf{B}}}{\bar{\mathbf{Q}}_x + \bar{\mathbf{B}}}; \quad \alpha_2 = \frac{(1 - \alpha_3)\bar{\mathbf{Q}}_x}{\bar{\mathbf{Q}}_x + \bar{\mathbf{B}}}; \quad \alpha_3 \in [0, 1]. \quad (3.10)$$

Применим предложенный алгоритм к приведенному примеру (3.7). Матрица \mathbf{Q} соответствует матрице \mathbf{Q}_x . Определим матрицы \mathbf{Q}_y как $\text{corr}(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2) = 0.2$, а все остальные элементы зададим 1. Рисунок 3.1 показывает значение векторов значимостей признаков \mathbf{z}_x и целевых векторов \mathbf{z}_y в зависимости от значения коэффициента α_3 . Если α_3 мало, значимости всех целевых векторов не различимы и значимость признака χ_3 выше значимости признака χ_2 . При увеличении α_3 до 0.2, коэффициент значимости $\mathbf{z}_{y,5}$ целевого вектора $\boldsymbol{\nu}_5$ увеличивается наряду со значимостью признака χ_2 .

3.2.3. Минимаксная постановка задачи выбора признаков

Функция (3.8) является симметричной по отношению к \mathbf{z}_x и \mathbf{z}_y . Она штрафует признаки, которые коррелированы и не имеют отношения к целевым векторам. Кроме того, она штрафует целевые векторы, которые коррелированы между собой и недостаточно коррелируют с признаками. Это приводит к малым значениям значимостей для целевых векторов, которые слабо коррелируют с признаками, и большим значениям для целевых векторов, которые сильно коррелируют с признаками. Этот результат противоречит интуиции. Цель —

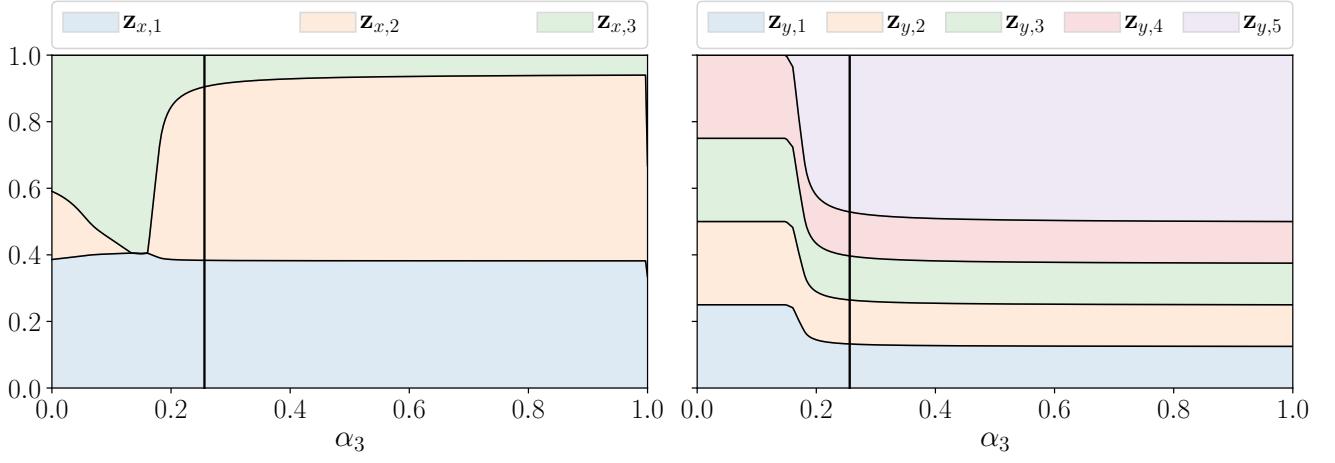


Рис. 3.1: Значимости признаков \mathbf{z}_x и целевых векторов \mathbf{z}_y в зависимости от α_3 для рассмотренного примера

предсказать все целевые векторы, особенно те, которые слабо коррелируют с признаками. Сформулируем две взаимосвязанные задачи:

$$\alpha_1 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x}_{\text{Sim}(\mathbf{X})} - \alpha_2 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y}_{\text{Rel}(\mathbf{X}, \mathbf{Y})} \rightarrow \min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n, \\ \mathbf{1}_n^\top \mathbf{z}_x = 1}} ; \quad (3.11)$$

$$\alpha_3 \cdot \underbrace{\mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y}_{\text{Sim}(\mathbf{Y})} + \alpha_2 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y}_{\text{Rel}(\mathbf{X}, \mathbf{Y})} \rightarrow \min_{\substack{\mathbf{z}_y \geq \mathbf{0}_r, \\ \mathbf{1}_r^\top \mathbf{z}_y = 1}} . \quad (3.12)$$

Разница между (3.11) и (3.12) заключается в знаке перед членом Rel. В пространстве входных объектов нерелевантные признаки должны иметь меньшие значения значимости. В то же время целевые вектора, не релевантные признакам, должны иметь большую значимость. Задачи (3.11) и (3.12) объединяются в совместную минимакс или максмин постановку

$$\min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n \\ \mathbf{1}_n^\top \mathbf{z}_x = 1}} \max_{\substack{\mathbf{z}_y \geq \mathbf{0}_r \\ \mathbf{1}_r^\top \mathbf{z}_y = 1}} f(\mathbf{z}_x, \mathbf{z}_y), \quad \left(\text{или } \max_{\substack{\mathbf{z}_y \geq \mathbf{0}_r \\ \mathbf{1}_r^\top \mathbf{z}_y = 1}} \min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n \\ \mathbf{1}_n^\top \mathbf{z}_x = 1}} f(\mathbf{z}_x, \mathbf{z}_y) \right), \quad (3.13)$$

где

$$f(\mathbf{z}_x, \mathbf{z}_y) = \alpha_1 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x}_{\text{Sim}(\mathbf{X})} - \alpha_2 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y}_{\text{Rel}(\mathbf{X}, \mathbf{Y})} - \alpha_3 \cdot \underbrace{\mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y}_{\text{Sim}(\mathbf{Y})} .$$

Теорема 3. Для положительно определенной матрицы \mathbf{Q}_x и \mathbf{Q}_y , максмин и минимакс задачи (3.13) имеют одинаковое оптимальное значение.

Доказательство. Введём обозначения

$$\mathbb{C}^n = \{\mathbf{z} : \mathbf{z} \geq \mathbf{0}_n, \mathbf{1}_n^\top \mathbf{z} = 1\}, \quad \mathbb{C}^r = \{\mathbf{z} : \mathbf{z} \geq \mathbf{0}_r, \mathbf{1}_r^\top \mathbf{z} = 1\}.$$

Множества \mathbb{C}^n и \mathbb{C}^r - компактные и выпуклые. Функция $f : \mathbb{C}^n \times \mathbb{C}^r \rightarrow \mathbb{R}$ является непрерывной. Если \mathbf{Q}_x и \mathbf{Q}_y положительно определены, функция f выпукло-вогнутая. Т. е., $f(\cdot, \mathbf{z}_y) : \mathbb{C}^n \rightarrow \mathbb{R}$ выпуклая при фиксированном \mathbf{z}_y , а $f(\mathbf{z}_x, \cdot) : \mathbb{C}^r \rightarrow \mathbb{R}$ вогнута при фиксированном \mathbf{z}_x . В этом случае по теореме Неймана о минимаксе

$$\min_{\mathbf{z}_x \in \mathbb{C}^n} \max_{\mathbf{z}_y \in \mathbb{C}^r} f(\mathbf{z}_x, \mathbf{z}_y) = \max_{\mathbf{z}_y \in \mathbb{C}^r} \min_{\mathbf{z}_x \in \mathbb{C}^n} f(\mathbf{z}_x, \mathbf{z}_y).$$

□

Для решения минимакс задачи (3.13), зафиксируем некоторый $\mathbf{z}_x \in \mathbb{C}^n$. Для фиксированного вектора \mathbf{z}_x решаем задачу

$$\max_{\mathbf{z}_y \in \mathbb{C}^r} f(\mathbf{z}_x, \mathbf{z}_y) = \max_{\substack{\mathbf{z}_y \geq \mathbf{0}_r \\ \mathbf{1}_r^\top \mathbf{z}_y = 1}} [\alpha_1 \cdot \mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y - \alpha_3 \cdot \mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y]. \quad (3.14)$$

Лагранжиан для данной задачи:

$$L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu}) = \alpha_1 \cdot \mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y - \alpha_3 \cdot \mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y + \lambda \cdot (\mathbf{1}_r^\top \mathbf{z}_y - 1) + \boldsymbol{\mu}^\top \mathbf{z}_y.$$

Здесь вектор множителей Лагранжа $\boldsymbol{\mu}$, который соответствует ограничениям на неравенства $\mathbf{z}_y \geq \mathbf{0}_r$, является неотрицательным. Двойственной задачей является

$$\min_{\lambda, \boldsymbol{\mu} \geq \mathbf{0}_r} g(\mathbf{z}_x, \lambda, \boldsymbol{\mu}) = \min_{\lambda, \boldsymbol{\mu} \geq \mathbf{0}_r} \left[\max_{\mathbf{z}_y \in \mathbb{C}^r} L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu}) \right]. \quad (3.15)$$

Для задачи квадратичного программирования (3.14) с положительно определенными матрицами \mathbf{Q}_x и \mathbf{Q}_y выполняются условия сильной двойственности. Таким образом, оптимальное значение (3.14) равно оптимальному значению (3.15). Это позволяет перейти от решения задачи (3.13) к решению задачи

$$\min_{\mathbf{z}_x \in \mathbb{C}^n, \lambda, \boldsymbol{\mu} \geq \mathbf{0}_r} g(\mathbf{z}_y, \lambda, \boldsymbol{\mu}). \quad (3.16)$$

Полагая градиент $\nabla_{\mathbf{z}_y} L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu})$ равным нулю, получим оптимальное значение \mathbf{z}_y :

$$\mathbf{z}_y = \frac{1}{2\alpha_3} \mathbf{Q}_y^{-1} \left(-\alpha_2 \cdot \mathbf{B}^\top \mathbf{z}_x + \lambda \cdot \mathbf{1}_r + \boldsymbol{\mu} \right). \quad (3.17)$$

Двойственная функция принимает вид

$$\begin{aligned} g(\mathbf{z}_x, \lambda, \boldsymbol{\mu}) &= \max_{\mathbf{z}_y \in \mathbb{R}^r} L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu}) = \mathbf{z}_x^\top \left(-\frac{\alpha_2^2}{4\alpha_3} \cdot \mathbf{B} \mathbf{Q}_y^{-1} \mathbf{B}^\top - \alpha_1 \cdot \mathbf{Q}_x \right) \mathbf{z}_x \\ &\quad - \frac{1}{4\alpha_3} \lambda^2 \cdot \mathbf{1}_r^\top \mathbf{Q}_y^{-1} \mathbf{1}_r - \frac{1}{4\alpha_3} \cdot \boldsymbol{\mu}^\top \mathbf{Q}_y^{-1} \boldsymbol{\mu} + \frac{\alpha_2}{2\alpha_3} \lambda \cdot \mathbf{1}_r^\top \mathbf{Q}_y^{-1} \mathbf{B}^\top \mathbf{z}_x \\ &\quad - \frac{1}{2\alpha_3} \lambda \cdot \mathbf{1}_r^\top \mathbf{Q}_y^{-1} \boldsymbol{\mu} + \frac{\alpha_2}{2\alpha_3} \cdot \boldsymbol{\mu}^\top \mathbf{Q}_y^{-1} \mathbf{B}^\top \mathbf{z}_x + \lambda. \end{aligned} \quad (3.18)$$

Тем самым задача (3.16) является квадратичной задачей с $n + r + 1$ переменными.

3.2.4. Несимметричный учёт значимостей признаков и целевых переменных

Естественным способом преодоления проблемы алгоритма SymImp является добавление штрафа для целевых векторов, которые коррелируют с признаками. Добавим линейный член $\mathbf{b}^\top \mathbf{z}_y$ в член $\text{Rel}(\mathbf{X}, \mathbf{Y})$ следующим образом:

$$\alpha_1 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x}_{\text{Sim}(\mathbf{X})} - \alpha_2 \cdot \underbrace{\left(\mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y - \mathbf{b}^\top \mathbf{z}_y \right)}_{\text{Rel}(\mathbf{X}, \mathbf{Y})} + \alpha_3 \cdot \underbrace{\mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y}_{\text{Sim}(\mathbf{Y})} \rightarrow \min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n, \mathbf{1}_n^\top \mathbf{z}_x = 1 \\ \mathbf{z}_y \geq \mathbf{0}_r, \mathbf{1}_r^\top \mathbf{z}_y = 1}}. \quad (3.19)$$

Утверждение 6. Пусть вектор \mathbf{b} равен

$$b_j = \max_{i=1, \dots, n} [\mathbf{B}]_{i,j}.$$

Тогда значение коэффициентов значимостей вектора \mathbf{z}_y будут неотрицательными в $\text{Rel}(\mathbf{X}, \mathbf{Y})$ для задачи (3.19).

Доказательство. Утверждение следует из факта

$$\sum_{i=1}^n z_i b_{ij} \leq \left(\sum_{i=1}^n z_i \right) \max_{i=1, \dots, n} b_{ij} = \max_{i=1, \dots, n} b_{ij},$$

где $z_i \geq 0$ и $\sum_{i=1}^n z_i = 1$. □

Следовательно, функция (3.19) штрафует в меньшей мере признаки, которые имеют отношение к целевым векторам, и целевые вектора, которые недостаточно коррелированы с признаками.

Утверждение 7. Баланс между членами $\text{Sim}(\mathbf{X})$, $\text{Rel}(\mathbf{X}, \mathbf{Y})$ и $\text{Rel}(\mathbf{X}, \mathbf{Y})$ для задачи (3.19) достигается при следующих коэффициентах:

$$\alpha_1 \propto \overline{\mathbf{Q}}_y (\overline{\mathbf{b}} - \overline{\mathbf{B}}); \quad \alpha_2 \propto \overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y; \quad \alpha_3 \propto \overline{\mathbf{Q}}_x \overline{\mathbf{B}}.$$

Доказательство. Необходимые значения α_1 , α_2 , и α_3 являются решением следующей системы уравнений:

$$\alpha_1 + \alpha_2 + \alpha_3 = 1; \tag{3.20}$$

$$\alpha_1 \overline{\mathbf{Q}}_x = \alpha_2 \overline{\mathbf{B}}; \tag{3.21}$$

$$\alpha_2 (\overline{\mathbf{b}} - \overline{\mathbf{B}}) = \alpha_3 \overline{\mathbf{Q}}_y. \tag{3.22}$$

Здесь, в (3.21) уравновешены $\text{Sim}(\mathbf{X})$ с первым слагаемым $\text{Rel}(\mathbf{X}, \mathbf{Y})$, а в (3.22) уравновешены $\text{Sim}(\mathbf{Y})$ с $\text{Rel}(\mathbf{X}, \mathbf{Y})$. \square

Утверждение 8. Для случая $r = 1$, предложенные функции (3.8), (3.13) и (3.19) совпадают с оригинальным алгоритмом QPFS (3.5).

Доказательство. Если r равно 1, то $\mathbf{Q}_y = q_y$ - скаляр, $\mathbf{z}_y = 1$ и $\mathbf{B} = \mathbf{b}$. Задачи (3.8), (3.13) и (3.19) принимают вид

$$\alpha_1 \cdot \mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^\top \mathbf{b} \rightarrow \min_{\mathbf{z}_x \geq \mathbf{0}_n, \mathbf{1}_n^\top \mathbf{z}_x = 1}.$$

При $\alpha = \frac{\alpha_2}{\alpha_1 + \alpha_2}$ последняя задача принимает вид (3.5). \square

Таблица 3.1 демонстрирует основные идеи и функции ошибок для каждого алгоритма. RelAgg является базовой стратегией и не учитывает корреляции в целевом пространстве. SymImp штрафует попарные корреляции между целевыми векторами. MinMax более чувствителен к целевым векторам, которые трудно предсказать. Стратегия AsymImp добавляет линейный член к функции SymImp, чтобы сделать вклад признаков и целевых векторов асимметричным.

Таблица 3.1: Обзор предлагаемых обобщений многомерного QPFS алгоритма

Алгоритм	Идея	Функция ошибки $S(\mathbf{z} \mathbf{X}, \mathbf{Y})$
RelAgg	$\min [\text{Sim}(\mathbf{X}) - \text{Rel}(\mathbf{X}, \mathbf{Y})]$	$\min_{\mathbf{z}_x} [(1 - \alpha) \cdot \mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x - \alpha \cdot \mathbf{z}_x^\top \mathbf{B} \mathbf{1}_r]$
SymImp	$\min [\text{Sim}(\mathbf{X}) - \text{Rel}(\mathbf{X}, \mathbf{Y}) + \text{Sim}(\mathbf{Y})]$	$\min_{\mathbf{z}_x, \mathbf{z}_y} [\alpha_1 \cdot \mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y + \alpha_3 \cdot \mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y]$
MinMax	$\min [\text{Sim}(\mathbf{X}) - \text{Rel}(\mathbf{X}, \mathbf{Y})]$ $\max [\text{Rel}(\mathbf{X}, \mathbf{Y}) + \text{Sim}(\mathbf{Y})]$	$\min_{\mathbf{z}_x} \max_{\mathbf{z}_y} [\alpha_1 \cdot \mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y - \alpha_3 \cdot \mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y]$
AsymImp	$\min [\text{Sim}(\mathbf{X}) - \text{Rel}(\mathbf{X}, \mathbf{Y})]$ $\max [\text{Rel}(\mathbf{X}, \mathbf{Y}) + \text{Sim}(\mathbf{Y})]$	$\min_{\mathbf{z}_x, \mathbf{z}_y} [\alpha_1 \cdot \mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot (\mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y - \mathbf{b}^\top \mathbf{z}_y) + \alpha_3 \cdot \mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y]$

3.3. Анализ методов учета значимостей целевых переменных

Для оценки предложенных алгоритмов выбора признаков, введём критерии оценки качества выбранного количества признаков. Определим коэффициент мультикорреляции как среднее значение коэффициента множественной корреляции следующим образом:

$$R^2 = \frac{1}{r} \text{tr} \left(\mathbf{C}^\top \mathbf{R}^{-1} \mathbf{C} \right); \quad \text{where } \mathbf{C} = [\text{corr}(\boldsymbol{\chi}_i, \boldsymbol{\nu}_j)]_{i=1, \dots, n, j=1, \dots, r}, \mathbf{R} = [\text{corr}(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j)]_{i,j=1}^n.$$

Этот коэффициент принимает значение между 0 и 1. Большее значение R^2 соответствует лучшему подмножеству признаков.

Нормированный среднеквадратичная ошибка (sRMSE) отображает качество прогнозирования модели. Оценка sRMSE считается на тренировочной и тестовой выборке.

$$\text{sRMSE}(\mathbf{Y}, \widehat{\mathbf{Y}}_{\mathbf{a}}) = \sqrt{\frac{\text{MSE}(\mathbf{Y}, \widehat{\mathbf{Y}}_{\mathbf{a}})}{\text{MSE}(\mathbf{Y}, \overline{\mathbf{Y}})}} = \frac{\|\mathbf{Y} - \widehat{\mathbf{Y}}_{\mathbf{a}}\|_2}{\|\mathbf{Y} - \overline{\mathbf{Y}}\|_2}.$$

Здесь $\widehat{\mathbf{Y}}_{\mathbf{a}} = \mathbf{X}_{\mathbf{a}} \boldsymbol{\Theta}_{\mathbf{a}}^\top$ – предсказание модель, $\overline{\mathbf{Y}}$ – предсказание константной модели, полученное усреднением целевой переменной по всем объектам. Данный

показатель на тестовой выборке необходимо минимизировать.

Байесовский информационный критерий (BIC) – компромисс между качеством предсказания и размером выбранного подмножества признаков $\|\mathbf{a}\|_0 = \#\{j : a_j \neq 0\} = \sum_{j=1}^n a_j$:

$$\text{BIC} = m \ln (\text{MSE}(\mathbf{Y}, \hat{\mathbf{Y}}_{\mathbf{a}})) + \|\mathbf{a}\|_0 \cdot \ln m,$$

Чем меньше значение BIC, тем лучше набор признаков.

Данные

Вычислительный эксперимент проводился на данных электрокортикограмм. Описание данных приведено в Главе 1.

На Рис. 3.2 показаны матрицы корреляций для исходных матриц \mathbf{X} и \mathbf{Y} данных ECoG. Частоты в матрице \mathbf{X} сильно коррелированы. В целевой матрице \mathbf{Y} корреляции между осями несущественны по сравнению с корреляциями между последовательными моментами времени и эти корреляции спадают со временем.

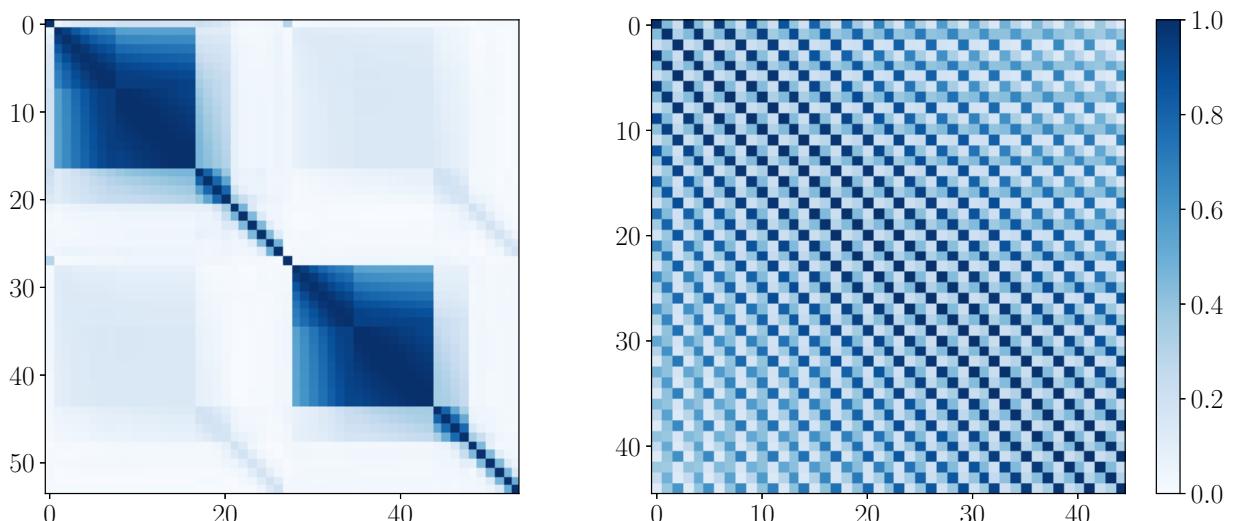


Рис. 3.2: Матрицы корреляций для матрицы плана \mathbf{X} и целевой матрицы \mathbf{Y} для данных ECoG

Результаты

Применим алгоритм SymImp QPFS для различных значений коэффициента α_3 согласно формуле (3.10). Зависимость значимости целевых векторов \mathbf{z}_y относительно коэффициента α_3 для различных значений k показана на Рис. 3.3. Значимости целевых векторов почти одинаковы для всех координат запястья при прогнозировании одного отсчёта времени ($k = 1$), что отражает независимость между координатами x , y и z . Для $k = 2$ и $k = 3$ значимости некоторых целевых векторов становится нулевой при увеличении α_3 . Вертикальные линии соответствуют оптимальному значению α_3 , вычисленному по (3.9). При этом значении α_3 значимости компонент \mathbf{z}_y совпадают. Таким образом, алгоритм не учитывает различия между целевыми векторами для $k = 1, 2, 3$.

Предлагаемые алгоритмы многомерного QPFS, приведенные в таблице 3.1 применяются для набора данных ECoG. Решим задачу выбора признаков для каждого из алгоритмов, чтобы получить вектора значимостей признаков. Отсортируем по убыванию признаки по значению их значимостей. Обучим линейную модель, постепенно добавляя в неё признаки. Исследуются значения описанных критериев качества при увеличении количества отобранных признаков. На Рис. 3.4 показаны результаты прогнозирования для случая прогнозирования $k = 30$ отсчётов времени. Порог значимости признаков τ обозначен цветными тиками. Пороговые значения τ для предлагаемых методов больше, чем для базового алгоритма RelAgg. Алгоритм SymImp имеет большой порог, не позволяя получить малый набор признаков. Однако алгоритм SymImp обладает наилучшей предсказательной способностью с точки зрения sRMSE на тестовых данных. Второй по качеству результат по sRMSE показал алгоритм AsymImp. Все предложенные алгоритмы достигают меньшей ошибки на тестовой выборке по сравнению с алгоритмом RelAgg. Критерий устойчивости также выше для предложенных алгоритмов. Алгоритм AsymImp показывает лучшие результаты с точки зрения качества прогнозирования и размера выбранного подмножества

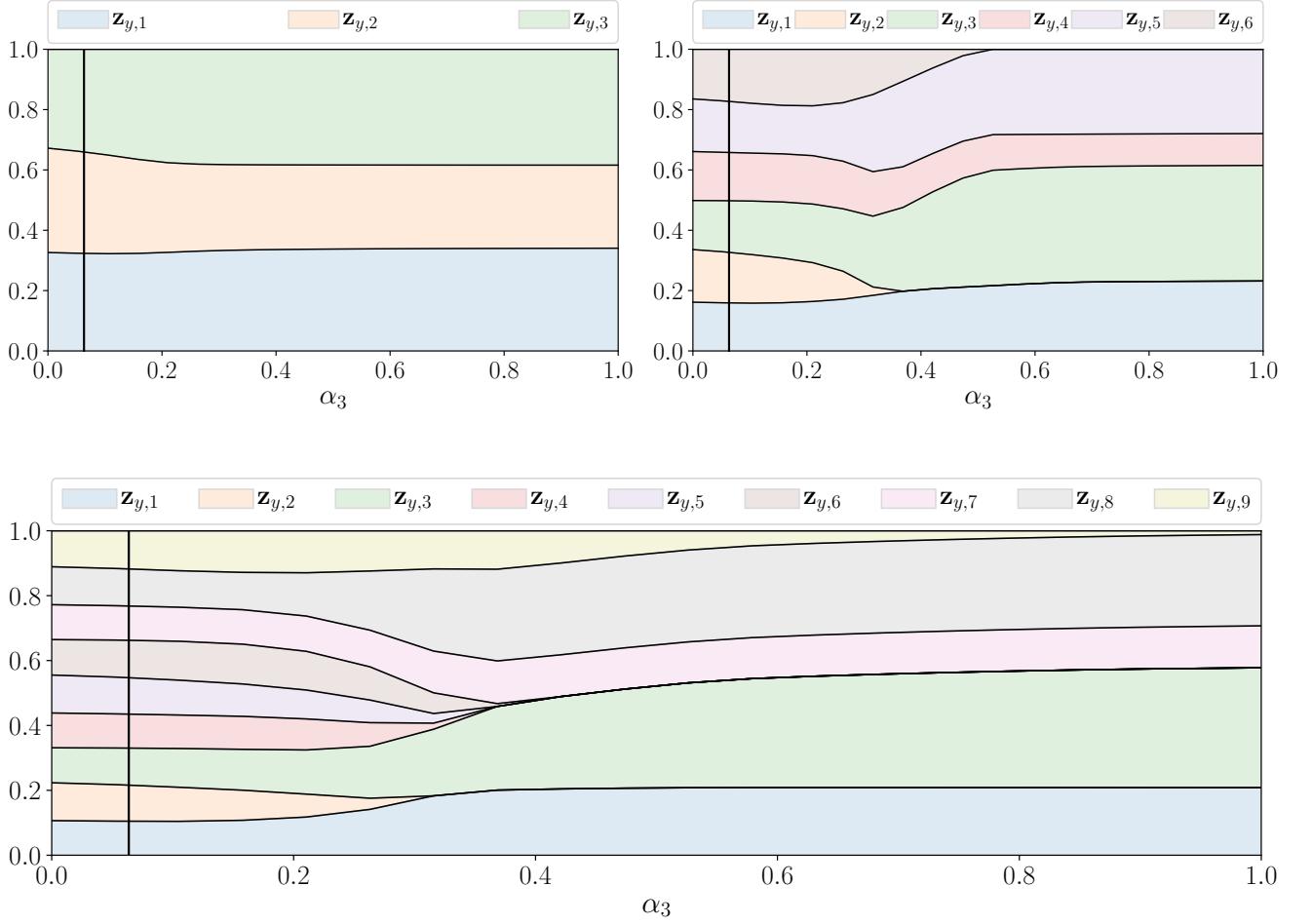


Рис. 3.3: Значимости целевых векторов \mathbf{Z}_y в зависимости от α_3 для алгоритма SymImp QPFS

признаков.

Чтобы сравнить структуру выбранных подмножеств признаков и исследовать стабильность процедуры выбора признаков, используется метод генерации данных с помощью бутстрепа. Генерируется множество подвыборок, выбирая объекты по одному с возвращениями. Затем решается задачу выбора признаков для каждой пары матрицы плана bX и целевой матрицы \mathbf{Y} . Сравниваются полученные вектора значимостей для различных подвыборок данных. В качестве меры стабильности работы алгоритмов вычисляется средний попарный коэффициент корреляции Спирмена и попарное ℓ_2 расстояние. В таблице 3.2 показана средняя ошибка sRMSE, размер подмножества признаков и описанные статистики для каждого алгоритма. Ошибка считалась на обученной линей-



Рис. 3.4: Сравнение предложенных алгоритмов выбора признаков для данных ECoG при прогнозировании $k = 30$ отсчётов времени

ной модели с использованием 50 признаков с наибольшими значениями значимостей. Asymimp дает наименьшую ошибку на тестовой выборке. Размер выбранных подмножеств объектов завышен при использовании порогового значения $\tau = 10^{-4}$. Оптимальное значение τ может быть подобрано с помощью процедуры кросс валидации.

Для того, чтобы сравнить методы снижения размерности и выбора признаков, используется модель PLS. На Рис. 3.5 показана ошибка sRMSE на тренировочной и тестовой выборках в зависимости от размерности скрытого пространства l . Ошибка на тестовой выборке достигает минимума при $l = 11$. Алгоритм PLS является более гибким подходом по сравнению с линейной моделью, построенной на подмножестве признаков, так как использует все исходные признаки. Это приводит к меньшей ошибке, но модель не является разреженной.

Таблица 3.2: Стабильность предложенных алгоритмов выбора признаков

	sRMSE	$\ \mathbf{a}\ _0$	Spearman ρ	ℓ_2
RelAgg	0.965 ± 0.002	26.8 ± 3.8	0.915 ± 0.016	0.145 ± 0.018
SymImp	0.961 ± 0.001	224.4 ± 9.0	0.910 ± 0.017	0.025 ± 0.002
MinMax	0.961 ± 0.002	101.0 ± 2.1	0.932 ± 0.009	0.059 ± 0.004
AsymImp	0.955 ± 0.001	85.8 ± 10.2	0.926 ± 0.011	0.078 ± 0.007

На рис. 3.6 проведено сравнение 3 моделей: линейной регрессии и регрессии PLS, построенной на 100 признаках QPFS, и регрессии PLS со всеми признаками. Линейная регрессия со всеми признаками не рассматривается, так как ее результаты близки к константному прогнозу. На рисунке также приведены результаты алгоритмов lasso и elastic net, которые широко используются для выбора признаков. В данном эксперименте использовался алгоритм AsymImp QPFS. Размерность скрытого пространства PLS $l = 15$. Результаты регрессии PLS значительно лучше, линейной регрессии с признаками QPFS. Это означает, что последняя модель не является достаточно гибкой. Тем не менее, лучший результат показывает модель PLS, построенная на признаках QPFS. Данная модель является разреженной, так как использует только 100 исходных признаков. Способность модели PLS находить оптимальное скрытое представление данных улучшает предсказательную способность модели.

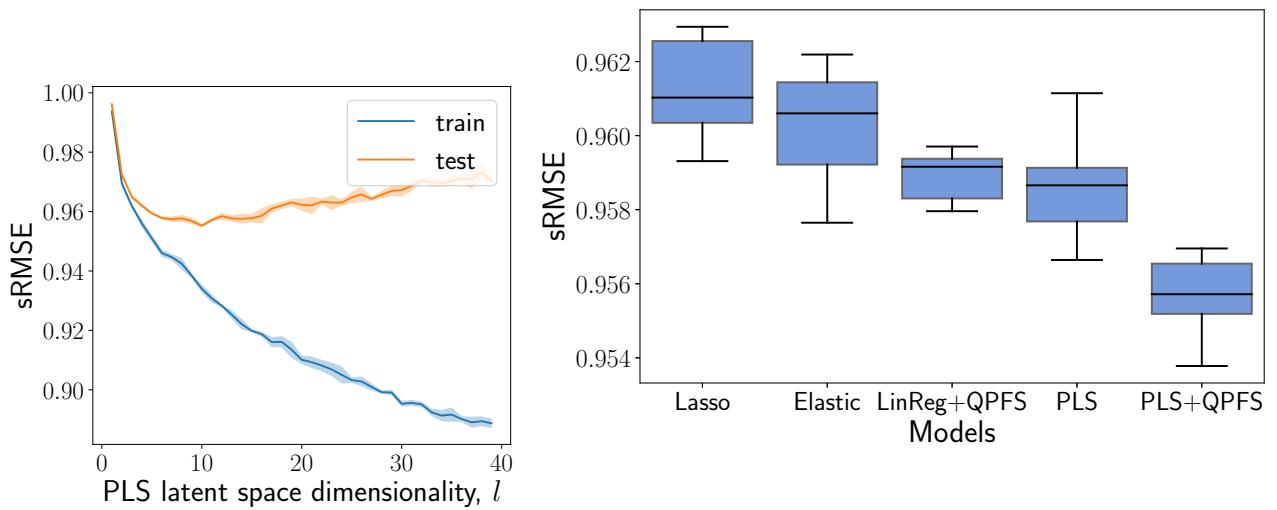


Рис. 3.6: Диаграммы размаха значений

Рис. 3.5: Ошибка sRMSE на тренировочной выборке и ошибка sRMSE на тестовой выборке для модели PLS для моделей Lasso, Elastic, LinReg+QPFS, PLS, PLS+QPFS

Глава 4

Выбор параметров нелинейных моделей с помощью квадратичного отбора признаков

4.1. Задача выбора параметров для обучения моделей

Модель $f(\mathbf{x}, \boldsymbol{\theta})$ с параметрами $\boldsymbol{\theta} \in \mathbb{R}^p$ предсказывает целевую переменную $y \in \mathbb{Y}$ по объекту $\mathbf{x} \in \mathbb{R}^n$. Пространство \mathbb{Y} представляет собой бинарные метки классов $\{0, 1\}$ для задачи двухклассовой классификации и \mathbb{R} для задачи регрессии. Даны матрица плана $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]^T \in \mathbb{R}^{m \times n}$ и целевой вектор $\mathbf{y} = [y_1, \dots, y_m]^T \in \mathbb{Y}^m$. Цель состоит в нахождении оптимальных параметров $\boldsymbol{\theta}^*$. Параметры $\boldsymbol{\theta}$ вычисляются минимизацией функции ошибки:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \mathcal{L}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}). \quad (4.1)$$

В качестве функции ошибки $\mathcal{L}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y})$ рассматриваются квадратичная ошибка для задачи регрессии:

$$\mathcal{L}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}) = \frac{1}{2} \|\mathbf{y} - \mathbf{f}(\mathbf{X}, \boldsymbol{\theta})\|_2^2 = \frac{1}{2} \sum_{i=1}^m (y_i - f(\mathbf{x}_i, \boldsymbol{\theta}))^2, \quad (4.2)$$

и функция кросс-энтропии для задачи бинарной классификации:

$$\mathcal{L}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}) = \sum_{i=1}^m [y_i \log f(\mathbf{x}_i, \boldsymbol{\theta}) + (1 - y_i) \log(1 - f(\mathbf{x}_i, \boldsymbol{\theta}))]. \quad (4.3)$$

Задача (4.1) решается с помощью итеративной процедуры оптимизации. Для получения параметров на шаге k текущие параметры $\boldsymbol{\theta}^{k-1}$ обновляются по следующему правилу:

$$\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1} + \Delta \boldsymbol{\theta}^{k-1}. \quad (4.4)$$

Авторы используют метод оптимизации Ньютона для выбора вектора обновлений $\Delta \boldsymbol{\theta}$.

Метод Ньютона нестабилен и вычислительно сложен. В данной статье предлагается стабильный алгоритм Ньютона. Перед шагом градиента предлагается

выбрать подмножество активных параметров модели, которые оказывают наибольшее влияние на функцию ошибки $\mathcal{L}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y})$. Обновление параметров производится только для отобранного множества индексов $\mathcal{A} = \{j : a_j = 1, \mathbf{a} \in \{0, 1\}^p\}$

$$\begin{aligned}\boldsymbol{\theta}_{\mathcal{A}}^k &= \boldsymbol{\theta}_{\mathcal{A}}^{k-1} + \Delta\boldsymbol{\theta}_{\mathcal{A}}^{k-1}, \quad \boldsymbol{\theta}_{\mathcal{A}} = \{\theta_j : j \in \mathcal{A}\}; \\ \boldsymbol{\theta}_{\bar{\mathcal{A}}}^k &= \mathbf{w}_{\bar{\mathcal{A}}}^{k-1}, \quad \boldsymbol{\theta}_{\bar{\mathcal{A}}} = \{\theta_j : j \notin \mathcal{A}\}.\end{aligned}$$

Чтобы выбрать оптимальное подмножество индексов \mathcal{A} , из всех возможных $2^p - 1$ подмножеств, вводится функция ошибки

$$\mathbf{a} = \arg \min_{\mathbf{a}' \in \{0,1\}^p} S(\mathbf{a}', \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}), \quad (4.5)$$

аналогичная функции ошибки (3.1) для задачи выбора признаков. Задача (4.5) решается на каждом шаге k процесса оптимизации для текущих параметров $\boldsymbol{\theta}^k$.

Алгоритм QPFS используется для решения задачи (4.5). QPFS выбирает подмножество параметров \mathbf{a} для вектора обновлений $\Delta\boldsymbol{\theta}$, которые оказывают наибольшее влияние на вектор остатков и являются попарно независимыми. Функция ошибки (3.5) соответствует функции ошибки $S(\mathbf{a}, \mathbf{X}, \mathbf{y}, \boldsymbol{\theta})$

$$\mathbf{a} = \arg \max_{\mathbf{a}' \in \{1,0\}^p} S(\mathbf{a}', \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) \Leftrightarrow \arg \min_{\mathbf{a} \in \mathbb{R}_+^p, \|\mathbf{a}\|_1=1} [\mathbf{a}^\top \mathbf{Q} \mathbf{a} - \alpha \cdot \mathbf{b}^\top \mathbf{a}]. \quad (4.6)$$

В работе показано, что для модели нелинейной регрессии с квадратичной функцией ошибки (4.2) и для модели логистической регрессии с кросс-энтропией (4.3), каждый шаг оптимизации эквивалентен задаче линейной регрессии (3.4).

4.2. Метод Ньютона решения задачи обучения параметров

Метод Ньютона использует условие оптимизации первого порядка для задачи (4.1) и линеаризует градиент $S(\boldsymbol{\theta})$

$$\nabla S(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}) = \nabla S(\boldsymbol{\theta}) + \mathbf{H} \cdot \Delta\boldsymbol{\theta} = 0,$$

$$\Delta\boldsymbol{\theta} = -\mathbf{H}^{-1}\nabla S(\boldsymbol{\theta}).$$

где $\mathbf{H} = \nabla^2 S(\boldsymbol{\theta})$ является Гессианом матрицы функции ошибки $S(\boldsymbol{\theta})$.

Итерация (4.4) метода Ньютона имеет вид

$$\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1} - \mathbf{H}^{-1}\nabla S(\boldsymbol{\theta}).$$

Каждая итерация инвертирует матрицу Гессиана. Мерой плохой обусловленности для матрицы Гессиана \mathbf{H} является число обусловленности

$$\varkappa(\mathbf{H}) = \frac{\lambda_{\max}(\mathbf{H})}{\lambda_{\min}(\mathbf{H})},$$

где $\lambda_{\max}(\mathbf{H}), \lambda_{\min}(\mathbf{H})$ являются максимальным и минимальным собственными значениями \mathbf{H} . Большое число обусловленности $\varkappa(\mathbf{H})$ приводит к нестабильности процесса оптимизации. Предложенный алгоритм уменьшает размер матрицы Гессиана \mathbf{H} . В наших экспериментах это приводит к меньшему числу обусловленности $\varkappa(\mathbf{H})$.

Размер шага метода Ньютона может быть чрезмерно большим. Для управления размером шага обновлений добавим параметр η в правило обновления (4.4)

$$\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1} + \eta\Delta\boldsymbol{\theta}^{k-1}, \quad \eta \in [0, 1].$$

Для выбора соответствующего размера шага η используется правило Армихо. Выбирается максимальное η так, чтобы выполнялось следующее условие

$$S(\boldsymbol{\theta}^{k-1} + \eta\Delta\boldsymbol{\theta}^{k-1}) < S(\boldsymbol{\theta}^{k-1}) + \gamma\eta\nabla S^T(\boldsymbol{\theta}^{k-1})\Delta\boldsymbol{\theta}^{k-1}, \quad \gamma \in [0, 0.5].$$

Модель нелинейной регрессии

Предположим, что модель $f(\mathbf{x}, \boldsymbol{\theta})$ близка к линейной в окрестности точки $\boldsymbol{\theta} + \Delta\boldsymbol{\theta}$

$$\mathbf{f}(\mathbf{X}, \boldsymbol{\theta} + \Delta\boldsymbol{\theta}) \approx \mathbf{f}(\mathbf{X}, \boldsymbol{\theta}) + \mathbf{J} \cdot \Delta\boldsymbol{\theta},$$

где $\mathbf{J} \in \mathbb{R}^{m \times p}$ является матрицы Якоби

$$\mathbf{J} = \begin{pmatrix} \frac{\partial f(\mathbf{x}_1, \boldsymbol{\theta})}{\partial \theta_1} & \cdots & \frac{\partial f(\mathbf{x}_1, \boldsymbol{\theta})}{\partial \theta_p} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(\mathbf{x}_m, \boldsymbol{\theta})}{\partial \theta_1} & \cdots & \frac{\partial f(\mathbf{x}_m, \boldsymbol{\theta})}{\partial \theta_p} \end{pmatrix}. \quad (4.7)$$

В соответствии с этим предположением градиент $\nabla S(\boldsymbol{\theta})$ и Гессиан матрицы \mathbf{H} функции ошибки (4.2) равняются

$$\nabla S(\boldsymbol{\theta}) = \mathbf{J}^\top (\mathbf{y} - \mathbf{f}), \quad \mathbf{H} = \mathbf{J}^\top \mathbf{J}. \quad (4.8)$$

Это приводит к методу Гаусса-Ньютона и правилу обновления (4.4)

$$\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1} + (\mathbf{J}^\top \mathbf{J})^{-1} \mathbf{J}^\top (\mathbf{f} - \mathbf{y}).$$

Вектор обновления $\Delta \boldsymbol{\theta}$ является решением задачи линейной регрессии

$$\|\mathbf{e} - \mathbf{F} \Delta \boldsymbol{\theta}\|_2^2 \rightarrow \min_{\Delta \boldsymbol{\theta} \in \mathbb{R}^p}, \quad (4.9)$$

где $\mathbf{e} = \mathbf{f} - \mathbf{y}$ и $\mathbf{F} = \mathbf{J}$.

В качестве нелинейной модели рассматривается модель двухслойной нейронной сети. В этом случае модель $f(\mathbf{x}, \boldsymbol{\theta})$ задается следующим образом:

$$f(\mathbf{x}, \boldsymbol{\theta}) = \sigma(\mathbf{x}^\top \mathbf{W}_1) \mathbf{w}_2.$$

Здесь $\mathbf{W}_1 \in \mathbb{R}^{m \times h}$ – это матрица весов, которые соединяют исходные признаки с h скрытыми нейронами. Функция нелинейности $\sigma(\cdot)$ применяется поэлементно. Веса $\mathbf{w}_2 \in \mathbb{R}^{h \times 1}$ соединяют скрытые нейроны с выходом. Вектор параметров модели $\boldsymbol{\theta}$ представляет собой объединение векторизованных матриц \mathbf{W}_1 , \mathbf{w}_2 .

Модель логистической регрессии

Для логистической регрессии модель имеет вид $f(\mathbf{x}, \boldsymbol{\theta}) = \sigma(\mathbf{x}^\top \boldsymbol{\theta})$ с сигмоидной функцией активации $\sigma(\cdot)$. Градиент и Гессиан функции ошибки (4.3) равны

$$\nabla S(\boldsymbol{\theta}) = \mathbf{X}^\top (\mathbf{f} - \mathbf{y}), \quad \mathbf{H} = \mathbf{X}^\top \mathbf{R} \mathbf{X}, \quad (4.10)$$

где \mathbf{R} – это диагональная матрица с диагональными элементами $f(\mathbf{x}_i, \boldsymbol{\theta}) \cdot (1 - f(\mathbf{x}_i, \boldsymbol{\theta}))$.

Правило обновления (4.4) в этом случае

$$\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1} + (\mathbf{X}^\top \mathbf{R} \mathbf{X})^{-1} \mathbf{X}^\top (\mathbf{y} - \mathbf{f}).$$

Этот алгоритм известен как итеративный алгоритм взвешенных наименьших квадратов (IRLS). Вектор обновлений $\Delta \mathbf{w}$ является решением задачи линейной регрессии

$$\|\mathbf{e} - \mathbf{F} \Delta \boldsymbol{\theta}\|_2^2 \rightarrow \min_{\Delta \boldsymbol{\theta} \in \mathbb{R}^p}, \quad (4.11)$$

где $\mathbf{e} = \mathbf{R}^{-1/2}(\mathbf{y} - \mathbf{f})$ и $\mathbf{F} = \mathbf{R}^{1/2} \mathbf{X}$.

4.3. Метод Ньютона с выбором параметром с помощью квадратичного программирования

Предлагается реализовать алгоритм QPFS для решения задач (4.9) и (4.11). QPFS матрица \mathbf{Q} и вектор \mathbf{b} имеют вид

$$\mathbf{Q} = \text{Sim}(\mathbf{F}), \quad \mathbf{b} = \text{Rel}(\mathbf{F}, \mathbf{e}).$$

Выборочный коэффициент корреляции равен нулю для ортогональных векторов. Покажем, что в оптимальной точке $\boldsymbol{\theta}^*$ вектор \mathbf{e} ортогонален столбцам матрицы \mathbf{F} . В этом случае вектор $\mathbf{b} = \text{Rel}(\mathbf{F}, \mathbf{e})$ равен нулю. Это означает, что член, учитывающий релевантность, в данном случае исключается. Условие оптимизации первого порядка гарантирует это свойство для модели нелинейной регрессии

$$\mathbf{F}^\top \mathbf{e} = \mathbf{J}^\top (\mathbf{f} - \mathbf{y}) = -\nabla S(\boldsymbol{\theta}^*) = \mathbf{0},$$

и для модели логистической регрессии

$$\mathbf{F}^\top \mathbf{e} = \mathbf{X} \mathbf{R}^{-1/2} \mathbf{R}^{1/2} (\mathbf{y} - \mathbf{f}) = \mathbf{X}^\top (\mathbf{y} - \mathbf{f}) = \nabla S(\boldsymbol{\theta}^*) = \mathbf{0}.$$

Псевдокод предлагаемого алгоритма приведён в алгоритме 2.

Algorithm 2 QPFS + Ньютон алгоритм

Вход: ε – допустимое отклонение;

τ – пороговое значение;

γ – параметр правила Армихо.

Выход: $\boldsymbol{\theta}^*$;

инициализировать $\boldsymbol{\theta}^0$;

$k := 1$;

повторять

вычислить \mathbf{e} и \mathbf{F} для (4.9) или (4.11) ;

$\mathbf{Q} := \text{Sim}(\mathbf{F})$, $\mathbf{b} := \text{Rel}(\mathbf{F}, \mathbf{e})$, $\alpha = \frac{\bar{\mathbf{Q}}}{\bar{\mathbf{Q}} + \bar{\mathbf{b}}}$;

$\mathbf{a} := \arg \min_{\mathbf{a} \geq 0, \|\mathbf{a}\|_1=1} \mathbf{a}^\top \mathbf{Q} \mathbf{a} - \alpha \cdot \mathbf{b}^\top \mathbf{a}$;

$\mathcal{A} := \{j : a_j = 1\}$;

вычислить $\nabla S(\boldsymbol{\theta}^{k-1})$, \mathbf{H} для (4.8) или (4.10);

$\Delta \boldsymbol{\theta}^{k-1} = -\mathbf{H}^{-1} \nabla S(\boldsymbol{\theta}^{k-1})$;

$\eta := \text{ArmijoRule}(\boldsymbol{\theta}^{k-1}, \gamma)$;

$\boldsymbol{\theta}_{\mathcal{A}}^k = \boldsymbol{\theta}_{\mathcal{A}}^{k-1} + \eta \Delta \boldsymbol{\theta}_{\mathcal{A}}^{k-1}$;

$k := k + 1$;

пока $\frac{\|\boldsymbol{\theta}^k - \boldsymbol{\theta}^{k-1}\|}{\|\boldsymbol{\theta}^k\|} < \varepsilon$

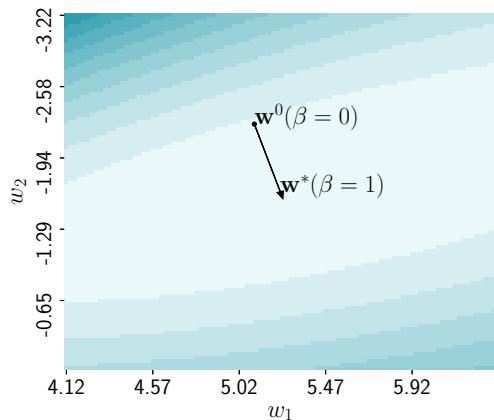


Рис. 4.1: Поверхность функции ошибки для логистической регрессии

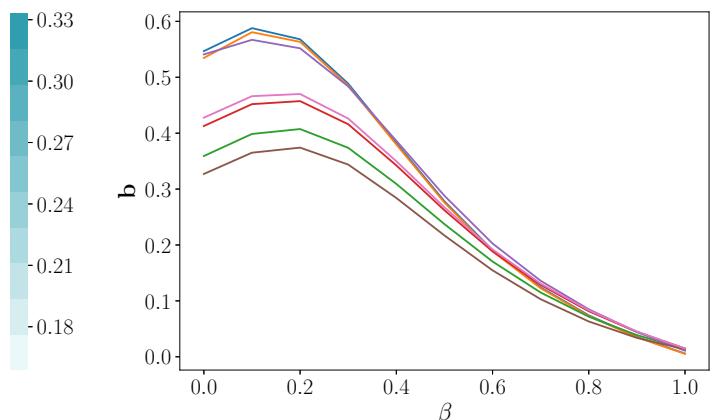


Рис. 4.2: Релевантность параметров для логистической регрессии

4.4. Анализ значимостей параметров нелинейных моделей

Целью вычислительного эксперимента является исследование свойств предложенного алгоритма и сравнение его с другими методами.

Исследована зависимость параметров алгоритма QPFS для задач (4.9), (4.11). Предположим, что вектор параметров θ^0 лежит вблизи оптимального вектора параметров θ^* . Рассмотрим отрезок

$$\theta_\beta = \beta\theta^* + (1 - \beta)\theta^0; \beta \in [0, 1].$$

Сгенерируем синтетический набор данных с 300 объектами и 7 признаками для задачи логистической регрессии. Ландшафт функции ошибки (4.3) на сетке двух случайно выбранных параметров показан на рис. 4.1. Поверхность функции ошибки выпуклая с вытянутыми линиями уровня вдоль некоторых параметров модели. Добавим случайный шум к оптимальным параметрам θ^* , чтобы получить точку θ^0 . Поведение вектора \mathbf{b} на отрезке между θ^0 и θ^* показано на рис. 4.2. Компоненты \mathbf{b} начинают резко уменьшаться, приближаясь к оптимальной точке.

Для модели нелинейной регрессии используется классический набор данных Boston Housing с 506 объектами и 13 признаками. Для простоты нейронная сеть

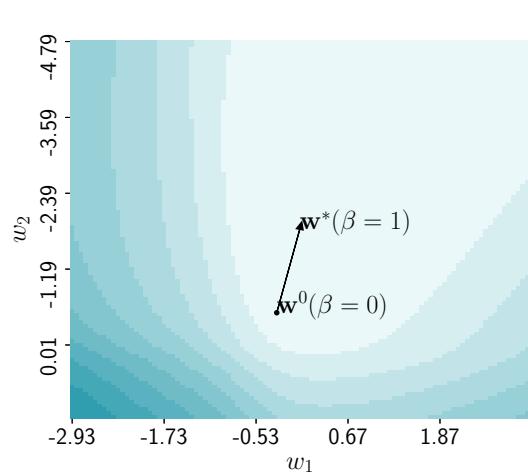


Рис. 4.3: Поверхность функции ошибки для нейронной сети

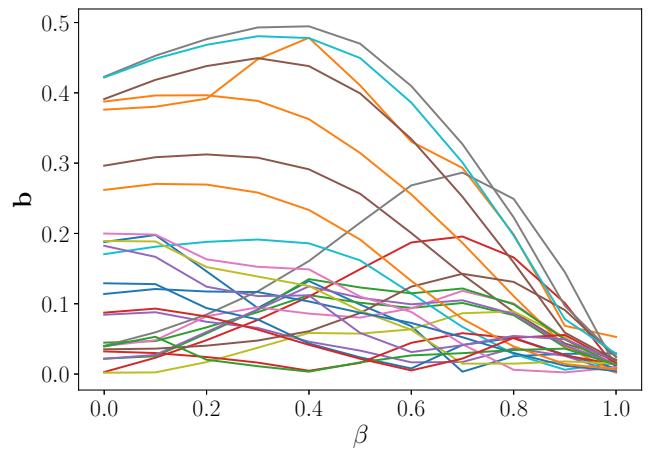


Рис. 4.4: Релевантность параметров первого слоя для модели нейронной сети

содержит два скрытых нейрона. Ландшафт функции ошибок для модели нейронной сети является более сложным. Он не выпуклый и может содержать несколько локальных минимумов. Двумерный ландшафт функции ошибок для этого набора данных показан на рис. 4.3. Сетка строится для двух случайных весов из матрицы \mathbf{W}_1 . Мы используем ту же стратегию для исследования того, как вектор \mathbf{b} изменяется от θ^0 до θ^* . Результат показан на рис. 4.4. Компоненты вектора \mathbf{b} становятся близки к нулю вблизи оптимума. При достижении оптимального значения различные веса влияют на остатки модели \mathbf{e} .

На рис. 4.5 показан процесс оптимизации для предложенного алгоритма в случае логистической регрессии с двумя параметрами модели. Даже для двумерной задачи решение метода Ньютона нестабильно и число обусловленности матрицы Гессиана \mathbf{H} может быть чрезвычайно большим. На каждом шаге алгоритма процедура QPFS выбирает параметры для оптимизации. В данном примере предложенный алгоритм выбирает и обновляет только один параметр на каждой итерации на первых шагах. Это делает алгоритм более устойчивым.

На рис. 4.6 показаны наборы активных параметров на итерациях для набора данных Boston Housing и нейронной сети с двумя скрытыми нейронами. Темные

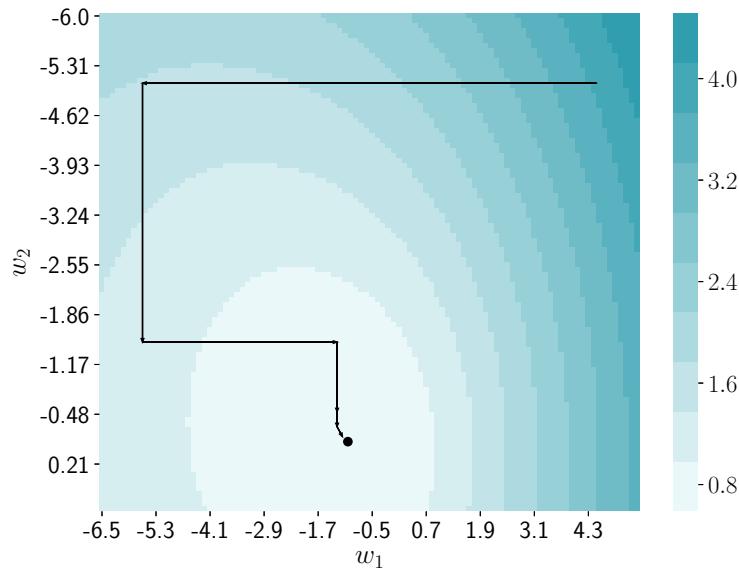


Рис. 4.5: Оптимизационный процесс предложенного алгоритма QPFS+Ньютон для модели логистической регрессии

ячейки соответствуют активным параметрам, которые мы оптимизируем.

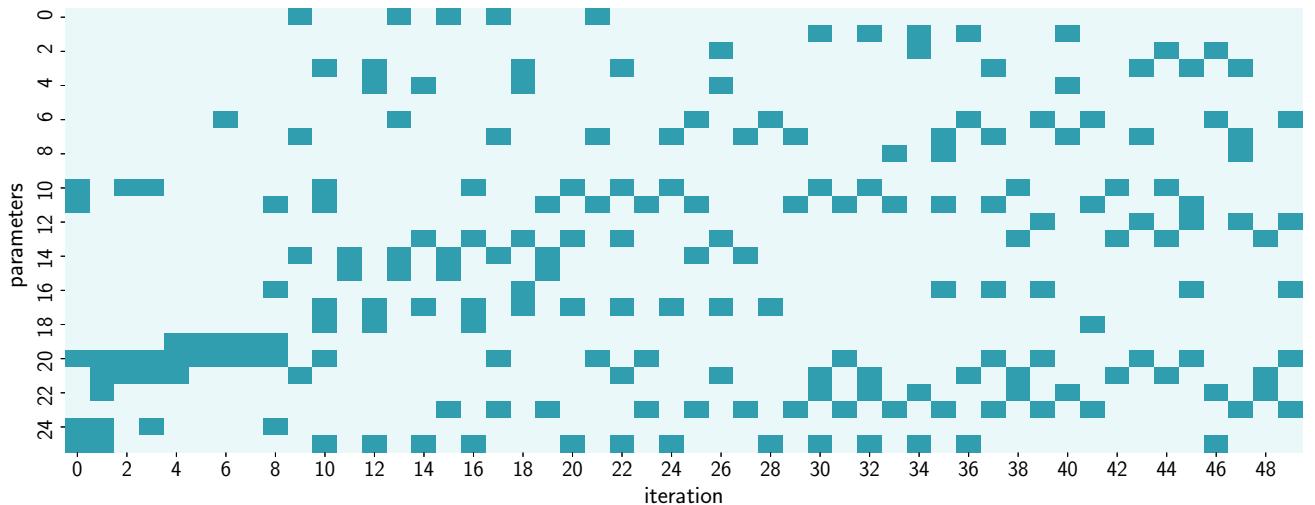


Рис. 4.6: Множество активных параметров на протяжении оптимизационного процесса

В рассмотренных примерах число обусловленности $\kappa(\mathbf{H})$ для метода Ньютона на некоторых итерациях было чрезвычайно большим. Выбор активных параметров позволил значительно сократить число обусловленности.

Мы сравнили предложенный алгоритм с существующими методами, а имен-

но градиентным спуском (GD), моментом Нестерова, Adam и оригинальным алгоритмом Ньютона. Проведены эксперименты для моделей нелинейной и логистической регрессий. Наборы данных были выбраны из репозитория UCI [85]. Результаты показаны в таблицах 4.1 и 4.2. Для каждого набора данных две строки содержат ошибки для тренировочной (первая строка) и тестовой (вторая строка) выборок. В таблице 4.1 приведена квадратичная ошибка, в таблице 4.2 – кросс-энтропия. Чтобы найти среднюю ошибку и ее стандартное отклонение использовалась процедура кросс валидации на 5 фолдов. Предложенный алгоритм показывает меньшую ошибку на трех из четырех наборов данных для нелинейной регрессии и среди двух из трех наборов данных для логистической регрессии.

Таблица 4.1: Средняя квадратичная ошибка на тренировочной и тестовой выборках для модели нелинейной регрессии

Выборка	mn	GD	Нестеров	ADAM	Ньютон	QPFS+Ньютон
Boston House Prices	506	27.2 ± 4.6	46.0 ± 11.0	35.4 ± 2.5	22.1 ± 15.2	20.9 ± 10.4
	13	32.4 ± 5.6	53.3 ± 11.5	37.8 ± 7.0	28.9 ± 13.6	24.5 ± 9.4
Communities and Crime	1994	48.0 ± 6.4	31.4 ± 2.8	23.3 ± 3.7	18.3 ± 3.4	26.7 ± 3.1
	99	47.5 ± 6.5	32.9 ± 4.3	28.1 ± 4.5	28.8 ± 3.6	28.4 ± 3.0
Forest Fires	517	18.9 ± 0.4	1.83 ± 0.4	1.81 ± 0.6	17.7 ± 0.4	17.9 ± 0.4
	10	20.0 ± 2.1	20.2 ± 2.2	20.0 ± 2.0	20.6 ± 1.4	20.2 ± 2.2
Residential Building	372	51.6 ± 17.7	32.6 ± 19.5	30.0 ± 24.8	35.5 ± 24.7	30.3 ± 10.7
	103	53.7 ± 13.9	34.1 ± 13.6	34.1 ± 19.4	35.0 ± 15.6	30.9 ± 5.3

Таблица 4.2: Среднее значение кросс-энтропии на тренировочной и тестовой выборках для модели логистической регрессии

Выборка	$\frac{m}{n}$	GD	Нестеров	ADAM	Ньютон	QPFS+Ньютон
Breast Cancer	569 30	0.6 ± 0.1 0.9 ± 0.2	0.4 ± 0.1 1.0 ± 0.7	0.8 ± 0.2 1.2 ± 0.2	0.3 ± 0.1 1.0 ± 0.2	0.2 ± 0.1 1.1 ± 0.3
Cardiotocography	2126 21	11.5 ± 4.7 11.6 ± 5.8	11.5 ± 4.7 11.5 ± 5.7	8.8 ± 4.4 9.0 ± 2.6	11.5 ± 5.7 11.5 ± 4.7	7.7 ± 4.2 7.7 ± 4.7
Climate Model Simulation Crashes	540 18	1.2 ± 0.1 1.4 ± 2.0	1.0 ± 0.2 1.3 ± 0.7	1.5 ± 0.2 1.8 ± 0.3	1.0 ± 0.5 1.2 ± 0.5	0.8 ± 0.3 1.1 ± 0.4

Глава 5

Метрические методы в задаче анализа временных рядов

связать все определения

5.1. Метрическое обучение в задачах кластеризации временных рядов

Пусть $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]^\top \in \mathbb{R}^{m \times n}$ — матрица плана. Объект $\mathbf{x}_i = [x_i^1, \dots, x_i^n]^\top$ задан в виде вектора в пространстве признаков. Требуется выявить кластерную структуру данных и разбить множество объектов \mathbf{X} на множество непересекающихся кластеров, т. е. построить отображение

$$f : \mathbf{X} \rightarrow \{1, \dots, r\}.$$

Обозначим $y_i = f(\mathbf{x}_i)$, $y_i \in \{1, \dots, r\}$, — метка кластера объекта \mathbf{x}_i . Необходимо выбрать метки кластеров $\{y_i\}_{i=1}^m$ таким образом, чтобы расстояния между кластерами были максимальными. Центроиды $\boldsymbol{\mu}$ множества объектов \mathbf{X} и центроиды кластеров $\{\boldsymbol{\mu}_j\}_{j=1}^r$ вычисляются по формулам:

$$\boldsymbol{\mu} = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i; \quad \boldsymbol{\mu}_j = \frac{\sum_{i=1}^m [y_i = y_j] \mathbf{x}_i}{\sum_{i=1}^m [y_i = y_j]}. \quad (5.1)$$

Введем на множестве объектов \mathbf{X} расстояние Махаланобиса

$$\rho_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A}^{-1} (\mathbf{x}_i - \mathbf{x}_j)}, \quad (5.2)$$

где \mathbf{A} — это матрица ковариаций множества \mathbf{X}

$$\mathbf{A} = \frac{1}{m} \sum_{i=1}^m (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top. \quad (5.3)$$

Определение 20. Функцией ошибки кластеризации назовем межклластерное расстояние:

$$\mathcal{L}(\{\boldsymbol{\mu}_j\}_{j=1}^r, \mathbf{X}, \mathbf{y}) = - \sum_{j=1}^r N_j \rho_{\mathbf{A}}^2(\boldsymbol{\mu}_j, \boldsymbol{\mu}), \quad (5.4)$$

где $N_j = \sum_{i=1}^m [y_i = y_j]$ — число объектов в кластере j .

Поставим задачу кластеризации как задачу минимизации функции ошибки (5.4)

$$(\{\boldsymbol{\mu}_j\}_{j=1}^r, \mathbf{X}, \mathbf{y}) \rightarrow \min_{\boldsymbol{\mu}_j \in \mathbb{R}^n} . \quad (5.5)$$

Для решения этой задачи предлагается применить метод метрического обучения к ковариационной матрице \mathbf{A} . Найдем такую матрицу \mathbf{A} , для которой функционал качества принимает максимальное значение:

$$\mathbf{A}^* = \arg \min_{\mathbf{A} \in \mathbb{R}^{n \times n}} S(\{\boldsymbol{\mu}_j^*\}_{j=1}^r, \mathbf{X}, \mathbf{y}), \quad (5.6)$$

где $\{\boldsymbol{\mu}_j^*\}_{j=1}^r$ — решение задачи кластеризации (5.5).

5.2. Алгоритм адаптивного метрического обучения

Для решения задач (5.5), (5.6) используется алгоритм адаптивного метрического обучения. Предлагается понизить размерность пространства объектов \mathbf{X} с помощью линейного ортогонального преобразования $\mathbf{P} \in \mathbb{R}^{n \times l}$, $\mathbf{P}^\top \mathbf{P} = \mathbf{I}$, где новая размерность $l < n$

$$\mathbf{t}_i = \mathbf{P} \mathbf{x}_i \in \mathbb{R}^l, \quad i = 1, \dots, m.$$

Центроид $\hat{\boldsymbol{\mu}}$ множества объектов $\{\mathbf{t}_i\}_{i=1}^m$ вычисляется по формуле (5.1). Расстояния между объектами вычисляются по формуле (5.2), где в качестве матрицы $\hat{\mathbf{A}}$ используется матрица ковариаций (5.3) множества объектов $\{\hat{\mathbf{x}}_i\}_{i=1}^m$

$$\hat{\mathbf{A}} = \frac{1}{m} \sum_{i=1}^m (\mathbf{t}_i - \hat{\boldsymbol{\mu}})(\mathbf{t}_i - \hat{\boldsymbol{\mu}})^\top = \frac{1}{m} \sum_{i=1}^m \mathbf{P}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top \mathbf{P}^\top = \mathbf{P} \mathbf{A} \mathbf{P}^\top.$$

Определение 21. Индикаторной матрицей назовем матрицу $\mathbf{Y} = \{y_{ij}\} \in \mathbb{R}^{m \times r}$, где

$$y_{ij} = \begin{cases} 1, & \text{если } f(\mathbf{x}_i) = y_j; \\ 0, & \text{если } f(\mathbf{x}_i) \neq y_j. \end{cases}$$

Определение 22. Взвешенной индикаторной матрицей назовем матрицу $\mathbf{L} = \mathbf{Y}(\mathbf{Y}^\top \mathbf{Y})^{-1/2} = \{l_{ij}\} \in \mathbb{R}^{m \times r}$, элементы которой равны:

$$l_{ij} = \begin{cases} \frac{1}{\sqrt{N_j}}, & \text{если } f(\mathbf{x}_i) = y_j; \\ 0, & \text{если } f(\mathbf{x}_i) \neq y_j. \end{cases}$$

Теорема 4. С использованием данных обозначений задача кластеризации (5.5) и задача метрического обучения (5.6) сводятся к общей задаче минимизации функции ошибки [86]

$$\mathcal{L} = -\frac{1}{m} \text{trace}(\mathbf{L}^\top \mathbf{X}^\top \mathbf{P}^\top \hat{\mathbf{A}}^{-1} \mathbf{P} \mathbf{X} \mathbf{L}) = -\frac{1}{m} \text{trace}(\mathbf{L}^\top \mathbf{X}^\top \mathbf{P}^\top (\mathbf{P} \mathbf{A} \mathbf{P}^\top)^{-1} \mathbf{P} \mathbf{X} \mathbf{L}) \rightarrow \min_{\mathbf{P}, \mathbf{L}}. \quad (5.7)$$

Для решения задачи (5.7) используется ЕМ алгоритм. На каждом шаге итеративно вычисляются текущие оптимальные значения матриц \mathbf{P} и \mathbf{L} . На E -шаге необходимо найти матрицу \mathbf{L} , которая является решением оптимизационной задачи (5.7) при фиксированной матрице \mathbf{P} . В качестве начального приближения получим взвешенную индикаторную матрицу \mathbf{L} с помощью алгоритма кластеризации k -средних с евклидовой метрикой. На M -шаге производится нахождение оптимального значения матрицы \mathbf{P} при фиксированной матрице \mathbf{L} . Алгоритм завершается при стабилизации функционала \mathcal{L} на последовательности итераций.

Алгоритм k -средних

В данной работе базовым алгоритмом для сравнения является алгоритм k -средних. На первом шаге алгоритм выбирает из множества \mathbf{X} случайным образом r объектов $\{\boldsymbol{\mu}_j\}_{j=1}^r$ — начальные центроиды кластеров. Для каждого объекта \mathbf{x}_i вычисляется расстояние (5.2) до каждого центроида кластера $\boldsymbol{\mu}_j$ с единичной матрицей трансформаций \mathbf{A} . Объект \mathbf{x}_i относится к кластеру, расстояние до которого оказалось наименьшим. Далее производится вычисление

новых центроидов кластеров по формуле (5.1). Алгоритм завершается, если значения центроидов кластеров стабилизируются.

Оптимизация матрицы \mathbf{P} с фиксированной матрицей \mathbf{L}

Для любых двух квадратных матриц \mathbf{A} и \mathbf{B} справедливо $\text{trace}(\mathbf{AB}) = \text{trace}(\mathbf{BA})$. Данное свойство позволяет переформулировать задачу (5.7) следующим образом:

$$\mathcal{L} = -\frac{1}{m} \text{trace}(\mathbf{L}^\top \mathbf{X}^\top \mathbf{P}^\top (\mathbf{P}\mathbf{A}\mathbf{P}^\top)^{-1} \mathbf{P}\mathbf{X}\mathbf{L}) = -\frac{1}{m} \text{trace}((\mathbf{P}\mathbf{A}\mathbf{P}^\top)^{-1} \mathbf{P}\mathbf{X}\mathbf{L}\mathbf{L}^\top \mathbf{X}^\top \mathbf{P}^\top).$$

Теорема 5. Обозначим $\mathbf{B} = \mathbf{X}\mathbf{L}\mathbf{L}^\top \mathbf{X}^\top$. Обозначим через $\mathbf{P} = [\mathbf{v}_1, \dots, \mathbf{v}_r]^\mathbf{T}$ матрицу, состоящую из r собственных векторов матрицы $\mathbf{A}^{-1}\mathbf{B}$, отвечающих наибольшим собственным значениям. Тогда решением (5.7) является ортогональная матрица, полученная QR -разложением матрицы \mathbf{P}^\top .

Доказательство. Функция ошибки \mathcal{L} зависит только от матрицы \mathbf{P} . Обозначим

$$s(\mathbf{P}) = \text{trace}((\mathbf{P}\mathbf{A}\mathbf{P}^\top)^{-1} \mathbf{P}\mathbf{B}\mathbf{P}^\top).$$

На данном шаге задача (5.7) принимает вид:

$$\mathbf{P}^* = \arg \max_{\mathbf{P} \in \mathbb{R}^{n \times l}} s(\mathbf{P}); \quad (5.8)$$

$$\mathbf{P}\mathbf{P}^\top = \mathbf{I}. \quad (5.9)$$

Ранг произведения матриц не превосходит рангов сомножителей, поэтому ранг матрицы \mathbf{B} не превосходит r . Решением (5.8) является матрица $\mathbf{P} = [\mathbf{v}_1, \dots, \mathbf{v}_r]^\mathbf{T}$, состоящая из r собственных векторов матрицы $\mathbf{A}^{-1}\mathbf{B}$, отвечающих наибольшим собственным значениям. Таким образом, размерность нового пространства объектов будет равна количеству кластеров r .

В общем случае матрица \mathbf{P} не является ортогональной. Заметим, что для любой невырожденной матрицы \mathbf{P} верно $s(\mathbf{P}) = s(\mathbf{MP})$. Для учета условия

ортогональности (5.9) найдем QR -разложение матрицы \mathbf{P} . Тогда ортогональная матрица \mathbf{Q} является оптимальным значением \mathbf{P}^* . \square

Оптимизация матрицы \mathbf{L} с фиксированной матрицей \mathbf{P}

Теорема 6. Обозначим $\hat{\mathbf{K}} = (1/N)\mathbf{X}^\top \mathbf{P}^\top \hat{\mathbf{A}}^{-1} \mathbf{P} \mathbf{X}$. Тогда задача (5.7) эквивалентна задаче кластеризации k -средних с заданным ядром $\hat{\mathbf{K}}$ [87].

При фиксированной матрице \mathbf{P} задача (5.7) принимает вид:

$$\text{trace}(\mathbf{L}^\top \hat{\mathbf{K}} \mathbf{L}) \rightarrow \max_{\mathbf{L} \in \mathbf{R}^{m \times r}} .$$

Матрица $\hat{\mathbf{K}}$ является симметричной и неотрицательно определенной, тем самым может быть выбрана в качестве ядра.

5.3. Постановка задачи метрического обучения с динамическим выравниванием временных рядов

Пусть объект $\mathbf{x}_i \in \mathbb{R}^n$ — временной ряд, последовательность измерений некоторой исследуемой величины в различные моменты времени. Пусть \mathbf{X} — множество всех временных рядов фиксированной длины n , $Y = \{1, \dots, K\}$ — множество меток классов. Пусть задана выборка $\mathfrak{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^\ell$ — множество объектов с известными метками классов $y_i \in Y$.

Требуется построить точную, простую, устойчивую модель классификации

$$a : \mathbf{X} \rightarrow Y.$$

Данную модель представим в виде суперпозиции

$$a(\mathbf{x}) = b \circ \mathbf{f} \circ G(\mathbf{x}, \{\mathbf{c}_e\}_{e=1}^K), \quad (5.10)$$

где G — процедура выравнивания временных рядов относительно центроидов классов $\{\mathbf{c}_e\}_{e=1}^K$, \mathbf{f} — алгоритм метрического обучения, b — алгоритм многоклассовой классификации.

5.3.1. Выравнивание временных рядов.

Для повышения качества и устойчивости алгоритма классификации предлагается провести выравнивание временных рядов каждого класса относительно центроида.

Пусть \mathbf{X}_e — множество объектов обучающей выборки \mathfrak{D} , принадлежащих одному классу $e \in \{1, \dots, K\}$. Центроидом множества объектов $\mathbf{X}_e = \{\mathbf{x}_i | y_i = e\}_{i=1}^{\ell}$ по расстоянию ρ назовем вектор $\mathbf{c}_e \in \mathbb{R}^n$ такой, что

$$\mathbf{c}_e = \operatorname{argmin}_{\mathbf{c} \in \mathbb{R}^n} \sum_{\mathbf{x}_i \in \mathbf{X}_e} \rho(\mathbf{x}_i, \mathbf{c}). \quad (5.11)$$

Для нахождения центроида предлагается в качестве расстояния между временными рядами использовать путь наименьшей стоимости [?], найденный методом динамической трансформации времени. Псевдокод решения оптимизационной задачи (5.11) приведен в алгоритме 3.

Algorithm 3 Нахождение центроида DBA(\mathbf{X}_e , n_iter)

Вход: \mathbf{X}_e — множество временных рядов, принадлежащих одному и тому же классу, n_iter — количество итераций алгоритма.

Выход: \mathbf{c} — центроид множества \mathbf{X}_e .

- 1: задать начальное приближение приближение центроида \mathbf{c} ;
- 2: **для** $i = 1, \dots, n_{\text{iter}}$
- 3: **для** $\mathbf{x} \in \mathbf{X}_e$
- 4: вычислить выравнивающий путь между \mathbf{c} и \mathbf{x}
 $\text{alignment}(\mathbf{x}) := \text{DTWalignment}(\mathbf{c}, \mathbf{x});$
- 5: объединить поэлементно множества индексов для каждого отсчета времени
 $\text{alignment} := \bigcup_{\mathbf{x} \in \mathbf{X}_e} \text{alignment}(\mathbf{x});$
- 6: $\mathbf{c} = \text{mean}(\text{alignment})$

DTWalignment(\mathbf{c}, \mathbf{x})

Вход: \mathbf{c}, \mathbf{x} — временные ряды.

Выход: alignment — выравнивающий путь. // каждый индекс временного ряда \mathbf{x} поставлен в однозначное соответствие индексу временного ряда \mathbf{c}

- 1: построить $n \times n$ -матрицу деформаций DTW
 $\text{cost} := \text{DTW}(\mathbf{c}, \mathbf{x});$
 - 2: вычислить выравнивающий путь по матрице деформаций
 $\text{alignment} := \text{DTWpath}(\text{cost});$
-

Общая процедура выравнивания имеет следующий вид:

- 1) построить множество центроидов классов $\{\mathbf{c}_e\}_{e=1}^K$;
- 2) по множеству центроидов найти пути наименьшей стоимости между каждым временным рядом \mathbf{x}_i и центроидом его класса \mathbf{c}_{y_i} ;
- 3) по каждому пути восстановить выравненный временной ряд;
- 4) привести множества выравненных временных рядов к нулевому среднему и нормировать на дисперсию.

Результатом выравнивания должно стать множество выравненных временных рядов.

5.3.2. Метрическое обучение.

Введем на множестве выравненных временных рядов расстояние Махаланобиса

$$d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{A} (\mathbf{x}_i - \mathbf{x}_j)},$$

где матрица трансформаций $\mathbf{A} \in \mathbb{R}^{n \times n}$ является симметричной и неотрицательно определенной ($\mathbf{A}^T = \mathbf{A}$, $\mathbf{A} \succeq 0$). Представим матрицу \mathbf{A} в виде разложения $\mathbf{A} = \mathbf{L}^T \mathbf{L}$. Матрица $\mathbf{L} \in \mathbb{R}^{p \times n}$ — матрица линейного преобразования, где p задает размерность преобразованного пространства. Если параметр $p < n$, то происходит снижение размерности признакового пространства.

Расстояние $d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j)$ есть евклидово расстояние между \mathbf{Lx}_i и \mathbf{Lx}_j :

$$d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{L}^T \mathbf{L} (\mathbf{x}_i - \mathbf{x}_j)} = \sqrt{(\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j))^T (\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j))} = \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|_2$$

В качестве алгоритма метрического обучения в данной работе был выбран алгоритм LMNN. Данный алгоритм сочетает в себе идеи метода k ближайших соседей. Первая идея заключается в минимизации расстояний между k ближайшими объектами, находящимися в одном классе. Запишем функционал качества в виде

$$Q_1(\mathbf{L}) = \sum_{j \sim i} \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|^2 \rightarrow \min_{\mathbf{L}},$$

где $j \rightsquigarrow i$ означает, что \mathbf{x}_j является одним из k ближайших соседей для \mathbf{x}_i . Вторая идея состоит в максимизации расстояния между каждым объектом и его объектами-нарушителями. Объектом-нарушителем для \mathbf{x}_i назовем объект \mathbf{x}_l такой, что

$$\|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_l)\|^2 \leq \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|^2 + 1, \quad \text{где } j \rightsquigarrow i. \quad (5.12)$$

Таким образом, необходимо минимизировать следующий функционал:

$$Q_2(\mathbf{L}) = \sum_{j \rightsquigarrow i} \sum_l (1 - y_{il}) [1 + \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|^2 - \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_l)\|^2]_+ \rightarrow \min_{\mathbf{L}},$$

где $y_{il} = 1$, если $y_i = y_l$, и $y_{il} = 0$ в противном случае. Положительная срезка позволяет штрафовать только те объекты, которые удовлетворяют условию (5.12).

Задача метрического обучения состоит в нахождении линейного преобразования $\mathbf{f}(\mathbf{x}) = \mathbf{L}\mathbf{x}$, то есть нахождении матрицы \mathbf{L} в виде решения оптимационной задачи

$$Q(\mathbf{L}) = \mu Q_1(\mathbf{L}) + (1 - \mu) Q_2(\mathbf{L}) \rightarrow \min_{\mathbf{L}}, \quad (5.13)$$

где $\mu \in (0, 1)$ — весовой параметр, определяющий вклад каждого из функционалов. Задача (5.13) представляет собой задачу полуопределенного программирования [88] и может быть решена существующими оптимизационными пакетами.

Описать игрушечный рисунок

5.3.3. Классификация выравненных временных рядов в метрике Махаланобиса.

Пусть $\mathbf{x} \in \mathbf{X}$ — неразмеченный временной ряд. Выравниваем временной ряд \mathbf{x} относительно всех центроидов классов

$$\hat{\mathbf{x}}_e = G(\mathbf{x}, \mathbf{c}_e), \quad \text{где } e = \{1, \dots, K\}.$$

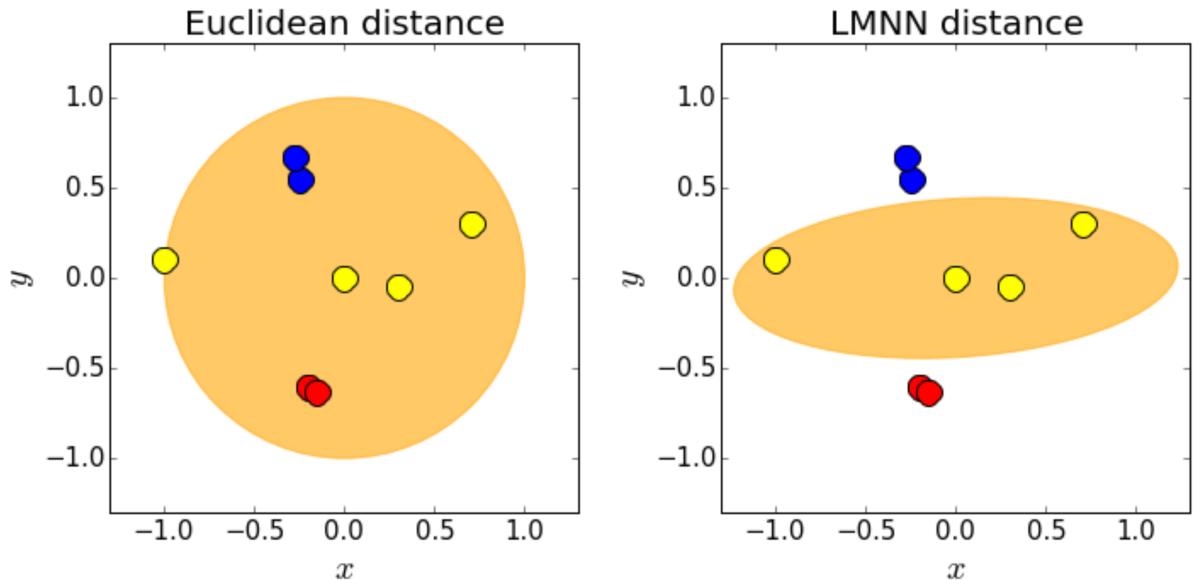


Рис. 5.1: Двумерный пример работы алгоритма LMNN

Отнесем временн \circ й ряд к классу, для которого минимально расстояние до со-ответствующего центроида. В качестве расстояния используем обученную метрику Махаланобиса с фиксированной матрицей \mathbf{A}

$$\hat{y} = \underset{e \in \{1, \dots, K\}}{\operatorname{argmin}} d_{\mathbf{A}}(\hat{\mathbf{x}}_e, \mathbf{c}_e).$$

После нахождения оптимальных центроидов классов и нахождения оптимальной матрицы трансформаций процедура классификации заключается в измерении расстояния между найденными центроидами и новыми неразмеченными объектами.

Для оценки качества работы алгоритма будем вычислять ошибку классификации как долю неправильно классифицированных объектов тестовой выборки \mathfrak{U} :

$$\text{error} = \frac{1}{|\mathfrak{U}|} \sum_{i=1}^{|\mathfrak{U}|} [a(\mathbf{x}_i) \neq y_i].$$

5.4. Анализ оптимального метрического пространства для задачи кластеризации

В целях проверки работоспособности предложенного подхода проведен вычислительный эксперимент на модельных данных. Сгенерирована выборка объектов, принадлежащих одному из двух классов, в двумерном пространстве. Каждый объект принадлежит многомерному нормальному распределению. На рис. 1 показано истинное распределение объектов, черным цветом выделены истинные центры классов и линии уровня функции распределения.

Применим к данной выборке базовый алгоритм k -средних. Результат кластеризации показан на рис. 2, где черным цветом выделены найденные центры классов и линии уровня функции распределения, построенной по выборочной ковариационной матрице.

Взяв за начальное приближение результаты работы алгоритма k -средних, проведем кластеризацию с помощью алгоритма адаптивного метрического обучения. Результаты работы алгоритма продемонстрированы на рис. 3.

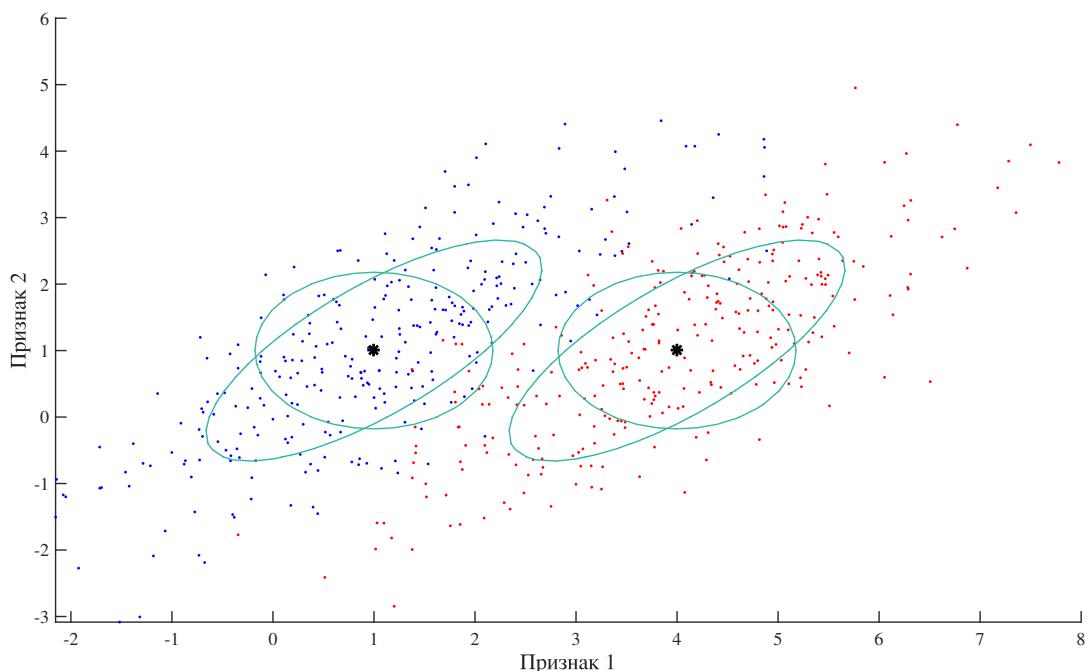


Рис. 5.2: Истинное распределение двумерных модельных данных

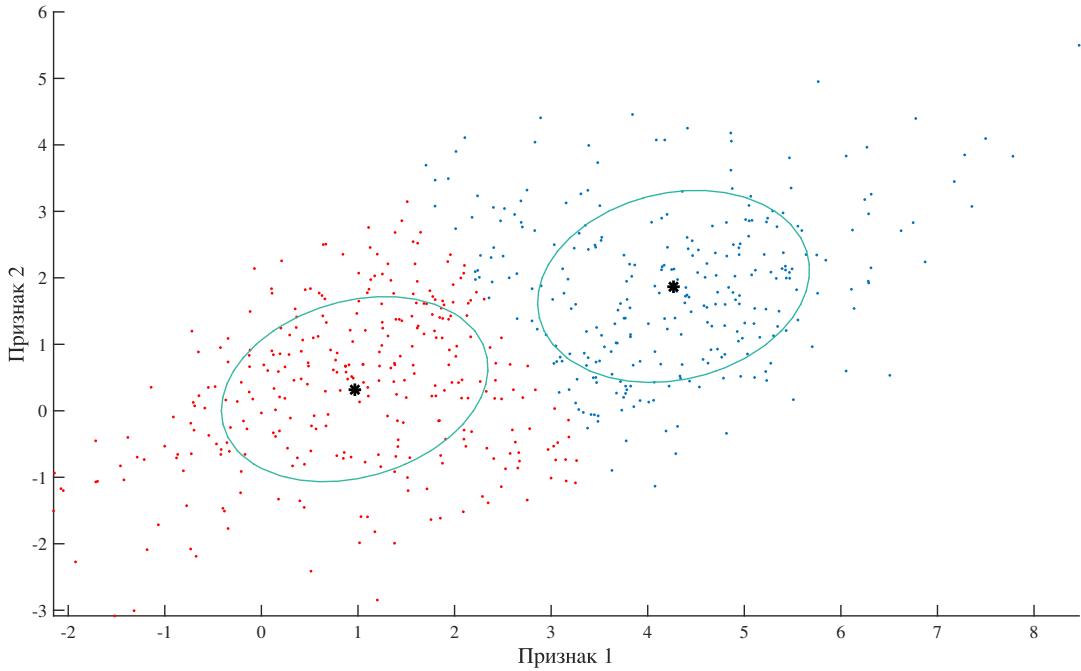


Рис. 5.3: Результат кластеризации модельных данных алгоритмом k -средних

На рисунках заметно улучшение результатов кластеризации. Измеренная точность кластеризации алгоритма k -средних составила 0,76, алгоритма адаптивного метрического обучения — 0,94, что говорит о работоспособности данного подхода.

Таблица 5.1 показывает результаты вычислительного эксперимента на реальных данных. Алгоритм был применен к 5 выборкам, взятых из репозитория UCI [85]. Оценкой качества кластеризации служит число правильно кластеризованных объектов. При кластеризации объектов на более чем два класса возникает проблема соотнесения истинных классов с полученными кластерами. Данная проблема была формализована в виде задачи о назначениях и решена с помощью венгерского алгоритма. Вычислительный эксперимент на реальных данных показал увеличение точности кластеризации при использовании метрического обучения.

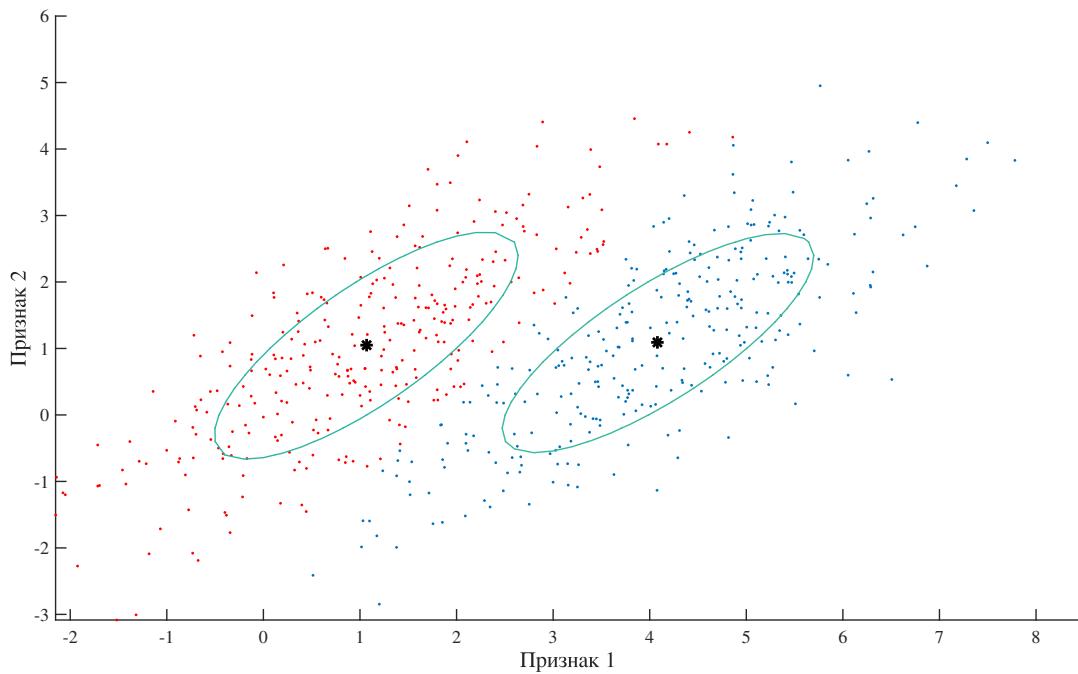


Рис. 5.4: Результат кластеризации модельных данных алгоритмом адаптивного метрического обучения

Таблица 5.1: Результаты кластеризации на множестве датасетов для методов k -средних и AML

Выборка	Качество кластеризации	
	k -средних	AML
Letter Recognition	0,356	0,428
Optical Recognition of Handwritten Digits	0,758	0,790
Seeds	0,833	0,881
Image Segmentation	0,545	0,737
Breast Cancer Wisconsin	0,960	0,956

5.5. Анализ метрического пространства для задачи классификации временных рядов

Цель вычислительного эксперимента — проверить работоспособность предложенного подхода. Предполагается, что построенный алгоритм мультиклассовой классификации способен определить тип активности человека по форме сигнала акселерометра мобильного телефона.

Для проведения базового вычислительного эксперимента были подготовлены синтетические временные ряды, принадлежащие двум классам. Первый класс — синусы вида $\sin(x + b)$, где параметр b определяет сдвиг каждого временного ряда. Второй класс — пилюобразные функции с различными сдвигами по временной шкале. На каждый временной ряд был наложен нормальный шум. Число временных рядов каждого класса = 60. Длина каждого временного ряда $n = 50$.

Построенные центроиды классов проиллюстрированы на рис. 5.5. Из рисунка видно, что процедура корректно определяет сдвиги временных рядов.

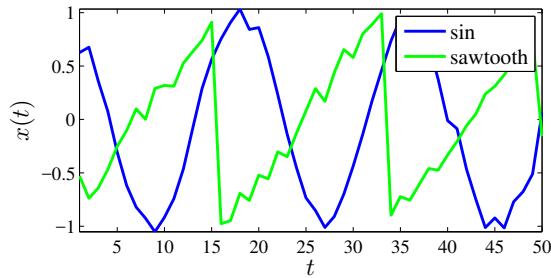


Рис. 5.5: Примеры центроидов синтетических временных рядов

Для того чтобы убедиться в целесообразности применения метрического обучения, данные временные ряды классифицировались в пространстве с евклидовой метрикой и в пространстве с метрикой Махalanобиса. Число ближайших соседей $k = 5$, размерность преобразованного пространства $p = 40$. Полученные ошибки классификации составили:

евклидова метрика — 27%

метрика Махalanобиса — 6%.

Реальные данные [89] представляли собой временные ряды акселерометра мобильного телефона. Каждый из шести классов соответствовал определенной физической активности испытуемых. Для проведения вычислительного эксперимента было выбрано по 200 объектов каждого класса. Длина каждого временного ряда равнялась $n = 128$ отсчетам времени.

Построенные центроиды классов изображены на рис. 5.6. Найденные центроиды обладают периодичностью, свойственной времененным рядам показаний активности человека. На рис. 5.7 показаны примеры временных рядов каждого

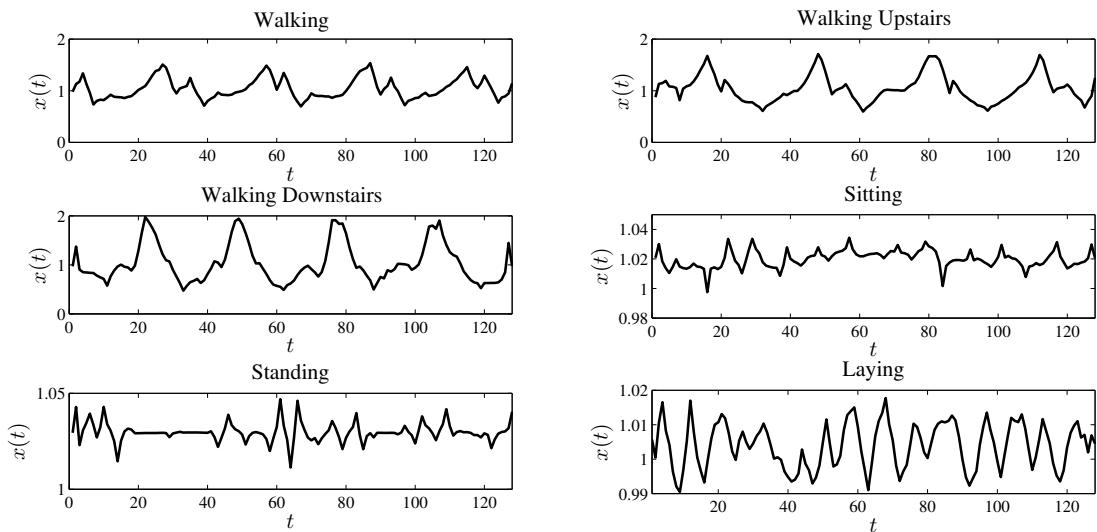


Рис. 5.6: Примеры центроидов временных рядов акселерометра

класса. Эти же временные ряды после процедуры выравнивания относительно построенных центроидов изображены на рис. 5.8.

Ошибка классификации без использования метрического обучения составила 37,5%. Алгоритм LMNN позволяет настроить параметры: число ближайших соседей k , размерность преобразованного евклидова пространства p . Для выбора оптимальных параметров воспользуемся процедурой кросс-проверки. На рис. 5.9 цветом показана ошибка классификации алгоритма в зависимости от его параметров. На данной выборке алгоритм LMNN оказывается слабо чув-

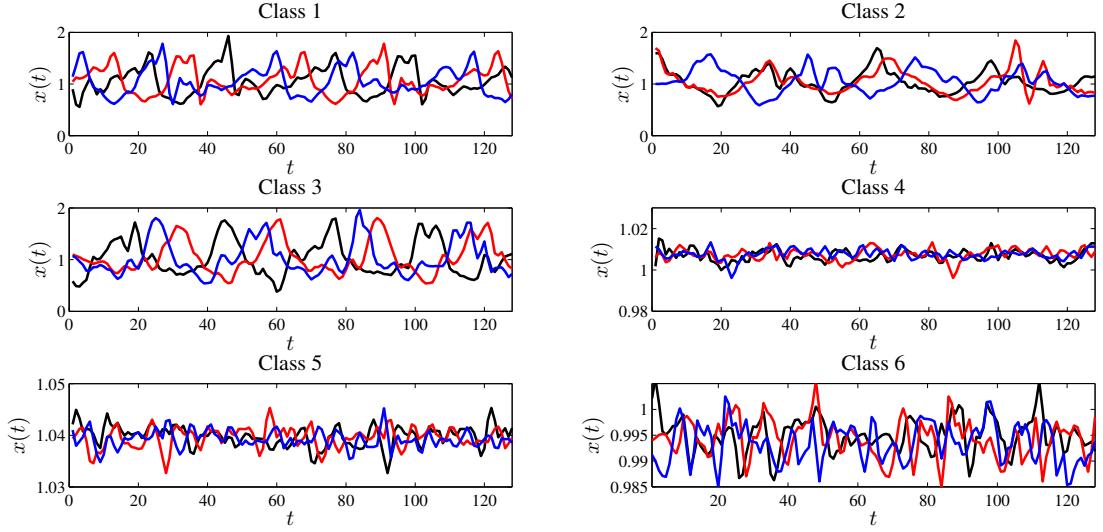


Рис. 5.7: Примеры временных рядов акселерометра

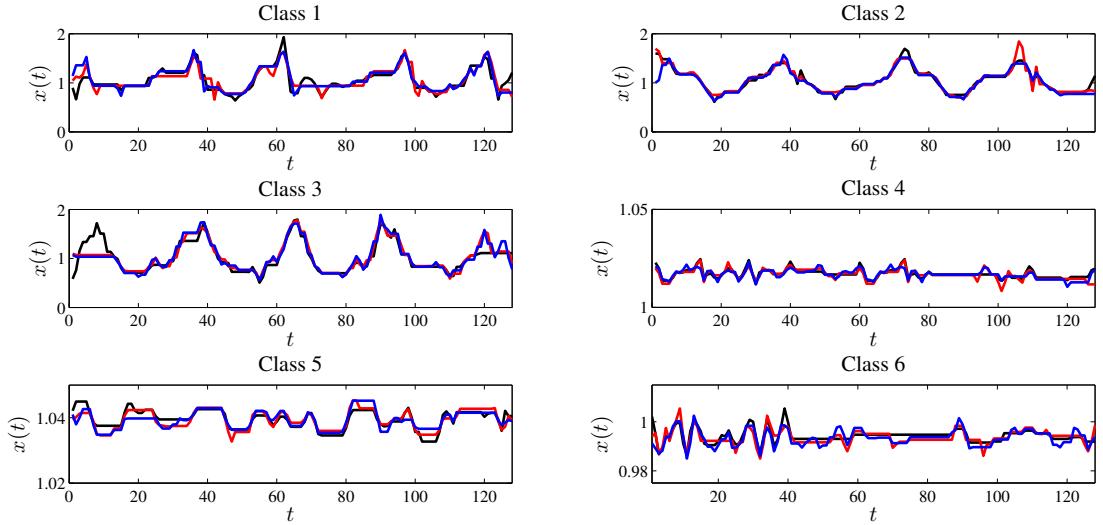


Рис. 5.8: Выравненные временные ряды акселерометра

ствителен к числу ближайших соседей, и при уменьшении размерности пространства объектов ошибка классификации растет.

Настроим алгоритм LMNN со следующими параметрами: число ближайших соседей $k = 30$, размерность выходного пространства $p = 128$. Ошибка классификации составила 17,25%, что вдвое меньше ошибки классификации с использованием евклидовой метрики.

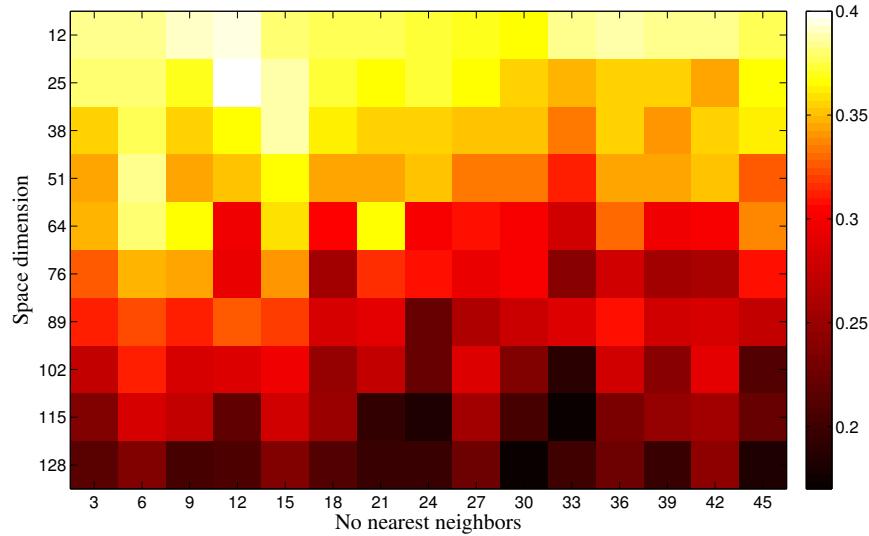


Рис. 5.9: Ошибка классификации метрического алгоритма в зависимости от размерности пространства и количества используемых ближайших соседей

Таблица 5.2: Матрицы несоответствий для Евклидовой метрики и метрики Махalanобиса, построенные для временных рядов акселерометра

		Истинные метки классов					
		1	2	3	4	5	6
(a) Евклидова метрика	1	80	0	5	0	0	0
	2	4	56	33	0	0	0
	3	5	5	86	0	0	0
	4	7	8	5	168	4	21
	5	51	61	57	12	192	11
	6	53	70	14	20	2	168

1	2	3	4	5	6		
(b) Метрика Махalanобиса	1	151	12	13	0	0	0
	2	10	142	14	0	0	0
	3	9	10	171	0	0	0
	4	10	7	0	173	9	21
	5	2	11	0	12	186	9
	6	18	18	2	15	5	170

В табл. 5.2 представлены матрицы несоответствий результатов классификации при использовании евклидовой метрики и метрики Махalanобиса. Столбцы соответствуют истинным меткам классов объектов, строки — предсказанным меткам. Диагональное преобладание матрицы несоответствий указывает

на высокую предсказательную способность алгоритма.

В табл. 5.3 продемонстрировано увеличение точности классификации при использовании в качестве меры расстояния метрики Махalanобиса. Пересечение i -го столбца и j -й строки отвечает изменению доли объектов класса i , относенных к классу j . Положительное суммарное значение диагональных элементов таблицы соответствует увеличению качества классификации. Значительное улучшение предсказания происходит при классификации первых трех классов. Данные классы соответствуют следующим видам физической активности: ходьба, ходьба вверх, ходьба вниз.

Таблица 5.3: Прирост точности классификации при использовании адекватной оценки матрицы трансформаций

	Истинные метки классов					
	1	2	3	4	5	6
1	0,355	0,06	0,04	0	0	0
2	0,03	0,43	-0,095	0	0	0
3	0,02	0,025	0,425	0	0	0
4	0,015	-0,005	-0,025	0,025	0,025	0
5	-0,245	-0,25	-0,28	0	-0,03	-0,01
6	-0,175	-0,26	-0,06	-0,025	0,005	-0,01

Глава 6

Порождение признаков с помощью метамоделей

6.1. Постановка задачи порождения признакового пространства

Временные ряды акселерометра образуют множество \mathcal{S} сегментов s фиксированной длины T :

$$s = [x_1, \dots, x_T]^T \in \mathbb{R}^T. \quad (6.1)$$

Необходимо построить модель классификации $f : \mathbb{R}^T \rightarrow Y$, которая будет ставить в соответствие каждому сегменту из множества \mathcal{S} метку класса из конечного множества Y . Обозначим за

$$\mathcal{D} = \{(s_i, y_i)\}_{i=1}^m \quad (6.2)$$

исходная выборка, где $s_i \in \mathcal{S}$ и $y_i = f(s_i) \in Y$.

Авторы предлагают построить модель f в виде суперпозиции $f = f(\mathbf{g})$. Функция $\mathbf{g} : \mathbb{R}^T \rightarrow \mathbb{R}^n$ является отображением из пространства \mathbb{R}^T в признаковое пространство $G \subset \mathbb{R}^n$. Имея функцию порождения признаков \mathbf{g} , преобразуем исходную выборку (6.2) в

$$\mathcal{D}_G = \{(\mathbf{g}_i, y_i)\}_{i=1}^m,$$

где $\mathbf{g}_i = \mathbf{g}(s_i) \in G$.

Модель классификации $f = f(\mathbf{g}, \theta)$ параметризована вектором θ . Оптимальные параметры $\hat{\theta}$ определяются оптимизацией функции ошибки классификации

$$\hat{\theta} = \arg \min_{\theta} L(\theta, \mathcal{D}_G, \mu). \quad (6.3)$$

Вектор μ является внешним параметром для заданной модели классификации. Примеры таких параметров и функций ошибки для различных моделей классификации приведены ниже.

Чтобы сравнить качество классификации с прошлыми результатами [90, 91], в качестве метрики качества используется точность классификации:

$$\text{accuracy} = \frac{1}{m} \sum_{i=1}^m \left[f \left(\mathbf{g}(s_i), \hat{\theta} \right) = y_i \right]. \quad (6.4)$$

6.2. Модели порождение признакового пространства для временных рядов

Цель данной работы — провести сравнение различных подходов к генерации признаков. В этом разделе проводится анализ рассматриваемых методов.

Экспертные функции. В качестве базового подхода будем использовать экспертные функции как функции порождения признаков. Экспертные функции — это некоторые статистики g_j , где $g_j : \mathbb{R}^T \rightarrow \mathbb{R}$. Признаком описанием $\mathbf{g}(s)$ объекта s являются значения заданных экспертных статистик для данного объекта

$$\mathbf{g}(s) = [g_1(s), \dots, g_n(s)]^\top.$$

В работе [92] авторы предлагают использовать экспертные функции, приведенные в таблице 6.1. Такая процедура порождения признаков генерирует признаковое описание временного ряда $\mathbf{g}(s) \in \mathbb{R}^{40}$.

Таблица 6.1: Примеры экспертных порождающих функций

Function description	Formula
Mean	$\bar{x} = \frac{1}{T} \sum_{t=1}^T x_t$
Standard deviation	$\sqrt{\frac{1}{T} \sum_{t=1}^T (x_t - \bar{x})^2}$
Mean absolute deviation	$\frac{1}{T} \sum_{t=1}^T x_t - \bar{x} $
Distribution	Histogram values with 10 bins

Авторегрессионная модель. Авторегрессионная модель [93] порядка n использует параметрическую модель для аппроксимации временного ряда s . Каждое значение временного ряда приближается линейной комбинацией предыдущих $n - 1$ значений

$$x_t = w_0 + \sum_{j=1}^{n-1} w_j x_{t-j} + \varepsilon_t,$$

где ε_t — регрессионные остатки. Оптимальные параметры $\hat{\mathbf{w}}$ авторегрессионной модели используются как признаки $\mathbf{g}(s)$. Данные параметры минимизируют квадратичную ошибку аппроксимации временного ряда и предсказания модели

$$\mathbf{g}(s) = \hat{\mathbf{w}} = \arg \min_{\mathbf{w} \in \mathbb{R}^n} \left(\sum_{t=n}^T \|x_t - \hat{x}_t\|^2 \right). \quad (6.5)$$

Задача (6.5) эквивалентна задаче линейной регрессии. Поэтому для каждого временного ряда s необходимо решить задачу линейной регрессии размера n . Пример аппроксимации временного ряда авторегрессионной моделью представлен на Рис. 6.1.

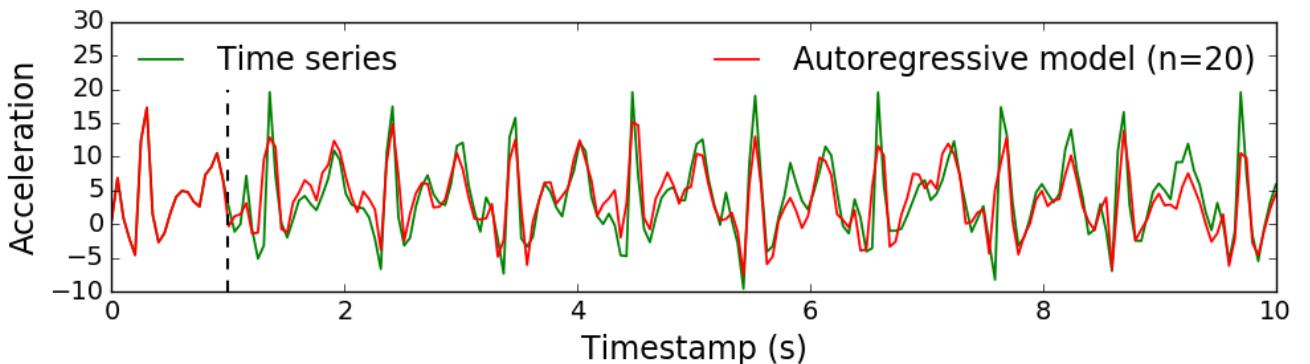


Рис. 6.1: Пример аппроксимации временного ряда авторегрессионной моделью с $n = 20$

Анализ сингулярного спектра. Альтернативной гипотезой порождения признакового пространства для временного ряда является анализ сингулярного спектра (Singular Spectrum Analysis, SSA) [94]. Для каждого временного

ряда s из выборки \mathcal{D} строится траекторная матрица:

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ x_2 & x_3 & \dots & x_{n+1} \\ \dots & \dots & \dots & \dots \\ x_{T-n+1} & x_{T-n+2} & \dots & x_T \end{pmatrix}.$$

Здесь ширина окна n является внешним структурным параметром. Сингулярное разложение матрицы $\mathbf{X}^\top \mathbf{X}$:

$$\mathbf{X}^\top \mathbf{X} = \mathbf{U} \Lambda \mathbf{U}^\top,$$

где \mathbf{U} — унитарная матрица и $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ причём λ_i собственные значения $\mathbf{X}^\top \mathbf{X}$. Признаковое описание объекта s задаётся спектром матрицы $\mathbf{X}^\top \mathbf{X}$:

$$\mathbf{g}(s) = [\lambda_1, \dots, \lambda_n]^\top.$$

Spline Approximation. Предлагаемый метод аппроксимирует временные ряды с помощью сплайнов [95]. Сплайн определяется его параметрами: узлами и коэффициентами. Предполагается, что узлы сплайна $\{\xi_\ell\}_{\ell=0}^M$ равномерно распределены по временной оси. Кусочные модели, построенные на отрезках $[\xi_{\ell-1}; \xi_\ell]$, заданы коэффициентами $\{\mathbf{w}_\ell\}_{\ell=1}^M$. Оптимальные параметры сплайна

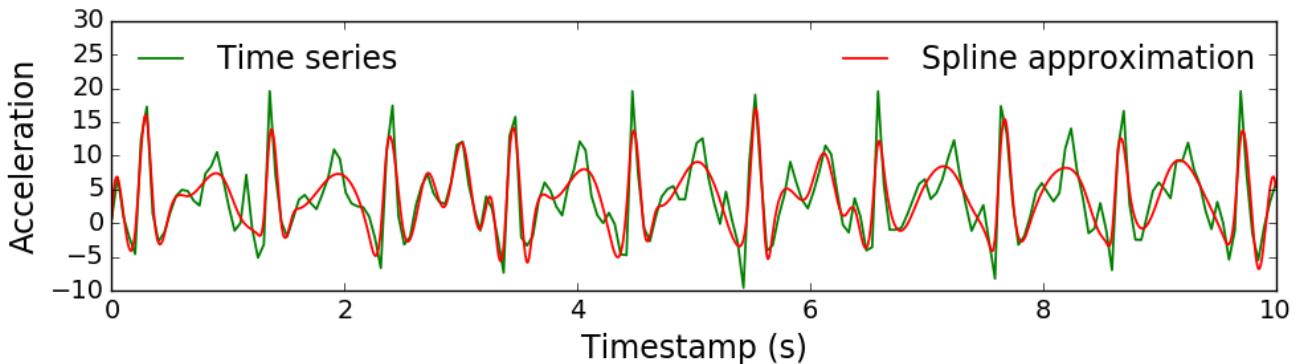


Рис. 6.2: Пример аппроксимации временного ряда с помощью сплайнов третьего порядка

являются решением системы с дополнительными условиями равенства производных до второго порядка включительно на концах отрезков. Обозначим каждый отрезок-сегмент $p_i(t)$ $i = 1, \dots, M$ и весь сплайн $S(t)$. Тогда система уравнений принимает вид

$$S(t) = \begin{cases} p_1(t) = w_{10} + w_{11}t + w_{12}t^2 + w_{13}t^3, & t \in [\xi_0, \xi_1], \\ p_2(t) = w_{20} + w_{21}t + w_{22}t^2 + w_{23}t^3, & t \in [\xi_1, \xi_2], \\ \dots & \dots \\ p_M(t) = w_{M0} + w_{M1}t + w_{M2}t^2 + w_{M3}t^3, & t \in [\xi_{M-1}, \xi_M], \end{cases}$$

$$\begin{aligned} S(\xi_t) &= x_t, \quad t = 0, \dots, M, \\ p'_i(\xi_i) &= p'_{i+1}(\xi_i), p''_i(\xi_i) = p''_{i+1}(\xi_i), \quad i = 1, \dots, M-1, \\ p_i(\xi_{i-1}) &= x_{i-1}, p_i(\xi_i) = x_i, \quad i = 1, \dots, M. \end{aligned}$$

Объединение всех параметров сплайна задаёт признаковое описание временного ряда:

$$\mathbf{g}(s) = [\mathbf{w}_1, \dots, \mathbf{w}_M]^\top.$$

Рис. 6.2 показывает аппроксимацию временного ряда с использованием модели сплайнов. По сравнению с авторегрессионной моделью сплайны строят более гладкую аппроксимацию, используя такое же количество параметров.

6.3. Классификация временных рядов в порожденном признаковом пространстве

Для классификации временных рядов будем использовать подход один против всех. Для каждого класса обучается бинарный классификатор, и на стадии предсказания объект классифицируется согласно наиболее уверенному классификатору. Использовались три модели классификации: логистическая регрессия, SVM и случайный лес.

Логистическая регрессия. Оптимальные параметры модели $\hat{\mathbf{w}}, \hat{b}$ в случае логистической регрессии определяются минимизацией функции ошибки (6.3)

$$L(\theta, \mathcal{D}_G, \mu) = \sum_{i=1}^m \log(1 + \exp(-y_i[\mathbf{w}^\top \mathbf{g}_i + b])) + \frac{\mu}{2} \|\mathbf{w}\|^2, \text{ where } \theta = \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}.$$

Решающее правило $f(\mathbf{g}, \theta)$ — знак линейной комбинации описания объекта \mathbf{g} и параметров $\hat{\theta}$

$$\hat{y} = f(\mathbf{g}, \hat{\theta}) = \operatorname{sgn}(\mathbf{g}^\top \hat{\mathbf{w}} + \hat{b}).$$

SVM. Оптимизационная задача метода SVM имеет вид

$$\hat{\theta} = \begin{pmatrix} \hat{\mathbf{w}} \\ \hat{b} \\ \hat{\xi} \end{pmatrix} = \arg \min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + \mu \sum_{i=1}^m \xi_i, \text{ s.t. } y_i (\mathbf{w}^\top \mathbf{g}_i + b) \geq 1 - \xi_i,$$

$$\xi_i \geq 0, \quad 1 \leq i \leq m.$$

Целевая функция соответствует функции ошибки классификации $L(\theta, \mathcal{D}_G, \mu)$. Предсказание для нового объекта вычисляется аналогично $\hat{y} = \operatorname{sgn}(\mathbf{g}^\top \hat{\mathbf{w}} + \hat{b})$.

Случайный лес. Случайный лес использует идею бэггинга. Идея состоит в построении многих слабых, неустойчивых классификаторов на подвыборках с возвращениями и усреднения их предсказаний. Метод предполагает использование в качестве базовых классификаторов моделей с низким смещением и высокой дисперсией. Усреднение позволяет уменьшить дисперсию. В случае случайного леса базовой моделью выступают решающие деревья. Идея бэггинга используется не только для самих объектов, но и для множества признаков. В данном случае предсказание для нового объекта получается усреднением всех предсказаний отдельных деревьев:

$$\hat{y} = \text{sgn} \left(\frac{1}{B} \sum_{i=1}^B \text{pred}(\mathbf{g}_i) \right),$$

где B — количество деревьев в композиции.

6.4. Анализ порожденных признаковых пространств

В данной работе эксперименты проводились на двух датасетах временных рядов акселерометра мобильного телефона: WISDM [89] и USC-HAD [96]. Акселерометр мобильного телефона проводит измерение ускорения по трём осям с частотой 100 Hz. Данные WISDM содержат 4321 временной ряд. Каждый временной ряд принадлежит к одному из 6 классов. Данные USC-HAD содержат 13620 временных рядов, принадлежащих одному из 12 классов. В таблице 6.2 представлено распределение временных рядов по классам для каждого датасета. Длина временного ряда равна 200. На рис. 6.3 представлен пример одного из временных рядов.

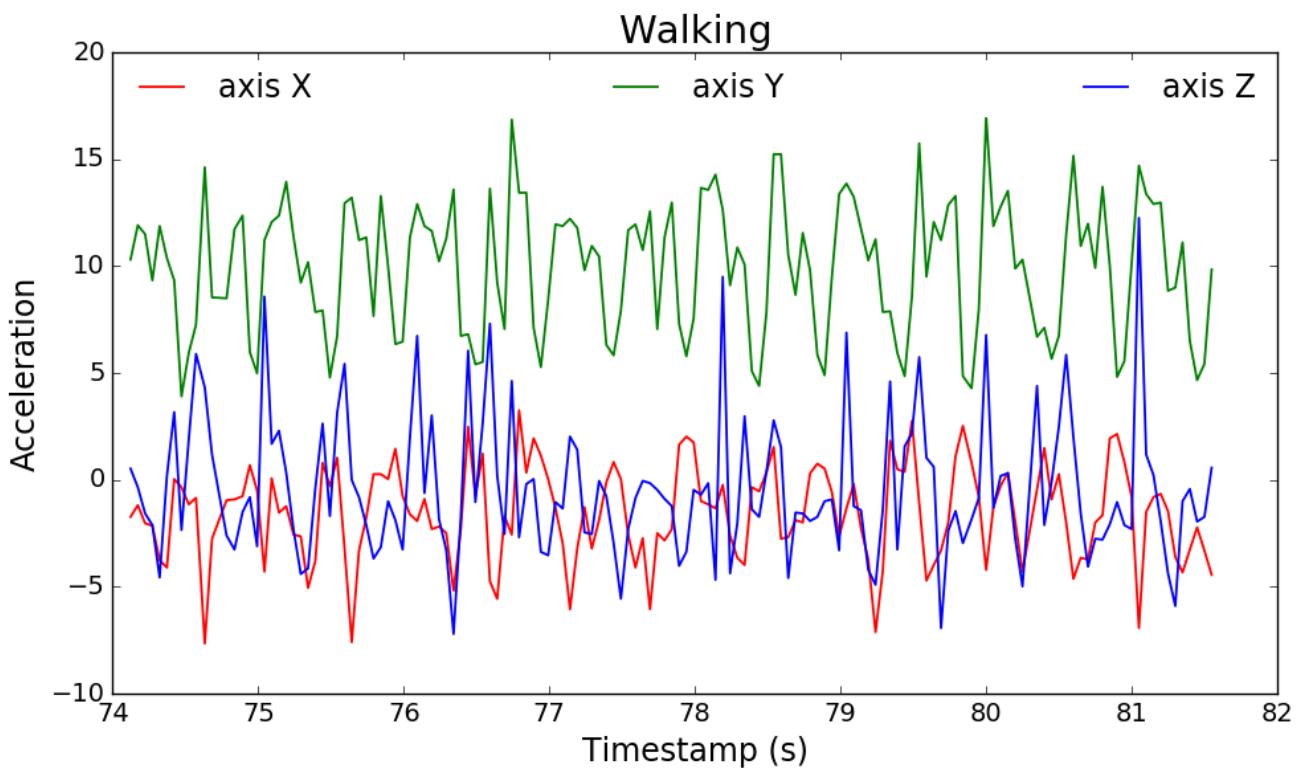


Рис. 6.3: Примеры временных рядов акселлерометра для каждой оси

Таблица 6.2: Распределение объектов по классам для временных рядов акселерометра

(a) WISDM			(b) USC-HAD		
	Activity	# objects		Activity	# objects
1	Standing	229 5.30 %	1	Standing	1167 8.57 %
2	Walking	1917 44.36 %	2	Elevator-up	764 5.61 %
3	Upstairs	466 10.78 %	3	Walking-forward	1874 13.76 %
4	Sitting	277 6.41 %	4	Sitting	1294 9.50 %
5	Jogging	1075 24.88 %	5	Walking-downstairs	951 6.98 %
6	Downstairs	357 8.26 %	6	Sleeping	1860 13.66 %
Total		4321	7	Elevator-down	763 5.60 %
			8	Walking-upstairs	1018 7.47 %
			9	Jumping	495 3.63 %
			10	Walking-right	1305 9.58 %
			11	Walking-left	1280 9.40 %
			12	Running	849 6.23 %
			Total		13620

В эксперименте для каждого датасета были порождены признаки одним из методов: экспертные функции, авторегрессионная модель, SSA и сплайны. Для каждой процедуры порождения признакового описания настраивались три модели классификации: логистическая регрессия, SVM и случайный лес. Внешние структурные параметры (длина авторегрессионной модели n , ширина окна SSA n , число узлов сплайна M) настраивались процедурой кросс-валидации:

$$CV(K) = \frac{1}{K} \sum_{k=1}^K L(f_k, \mathcal{D} \setminus \mathcal{C}_k), \quad (6.6)$$

где $C_k = \frac{K-1}{K}$ доля от всей выборки, используемая для обучения модели f_k .

Гиперпараметры μ моделей классификации были настроены той же процедурой

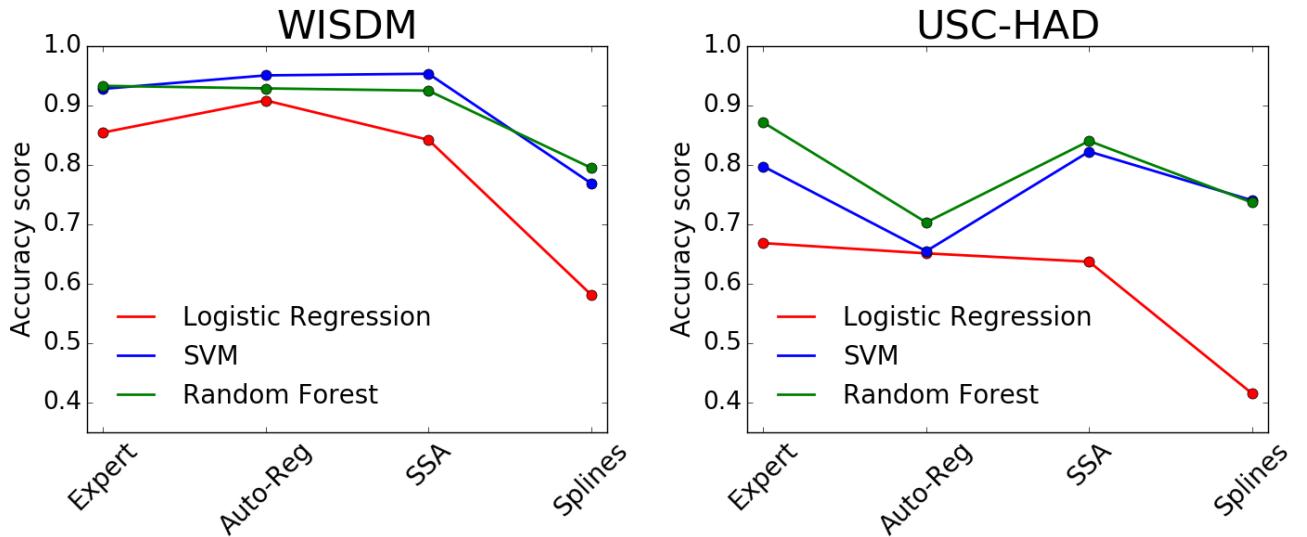


Рис. 6.4: Мультиклассовая точность классификации для различных порожденных признаковых пространств

кросс-валидации.

Первый подход к порождению признаков временных рядов — экспертные функции. Основной недостаток такого подхода необходимость экспертного задания функций и возможности вычисления их для конкретного датасета.

Авторегрессионная модель требует задания параметра длины модели n . Процедура кросс-валидации дала наибольшее качество при $n = 20$ для обоих датасетов.

Модель SSA была настроена аналошгичной процедурой выбора оптимальных гиперпараметров. Конечная модель имела ширину окна $n = 20$.

Для аппроксимации временных рядов кубическими сплайнами [95] использовалась библиотека *scipy*. Узлы сплайнов $\{\xi_\ell\}_{\ell=1}^M$ были распределены равномерно по временной оси. Значение параметра M было подобрано на кросс-валидации.

Для обоих датасетов процедуры порождения признаковых описаний дали следующие количества признаков: экспертные функции — 40; авторегрессионная модель — 60; анализ сингулярного спектра — 60; сплайны — 33.

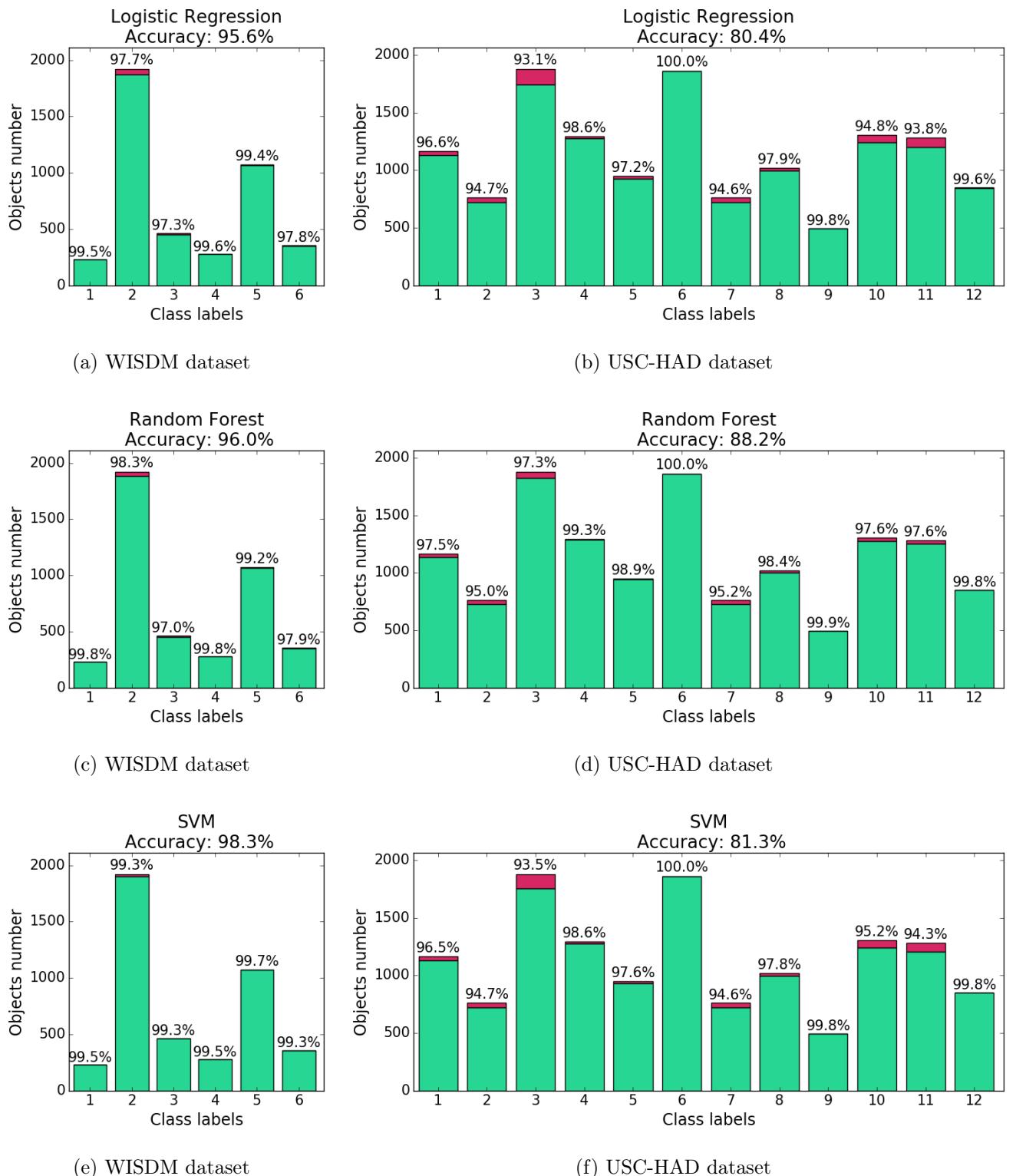


Рис. 6.5: Поклассовая точность классификации временных рядов акселерометра

На рис. 6.4 показано качество классификации (6.4) для двух датасетов. Для данных WISDM сплайны дали самое слабое качество классификации. Результаты для экспертных функций, авторегрессионной модели и SSA схожи. Для

Таблица 6.3: Бинарная точность классификации для данных WISDM с использованием рассматриваемых алгоритмов: EX — Expert, AR — Auto-Reg, SSA and SPL for Splines

	Logistic Regression				Random Forest				SVM			
	EX	AR	SSA	SPL	EX	AR	SSA	SPL	EX	AR	SSA	SPL
All	0.85	0.91	0.84	0.58	0.93	0.93	0.92	0.79	0.93	0.95	0.95	0.77
Standing	0.99	0.98	1.00	0.95	1.00	0.99	1.00	0.99	0.99	0.98	1.00	0.96
Walking	0.91	0.96	0.86	0.61	0.96	0.97	0.95	0.86	0.96	0.98	0.98	0.84
Upstairs	0.91	0.95	0.91	0.89	0.96	0.96	0.96	0.90	0.96	0.98	0.97	0.89
Sitting	0.99	0.98	1.00	0.99	1.00	0.99	1.00	1.00	0.99	0.98	1.00	1.00
Jogging	0.98	0.99	0.99	0.80	0.99	0.99	0.99	0.92	0.99	0.99	0.99	0.93
Downstairs	0.93	0.96	0.94	0.92	0.96	0.97	0.96	0.92	0.96	0.98	0.97	0.92

Таблица 6.4: Бинарная точность классификации для данных USC-HAD с использованием рассматриваемых алгоритмов: EX — Expert, AR — Auto-Reg, SSA and SPL for Splines

	Logistic Regression				Random Forest				SVM			
	EX	AR	SSA	SPL	EX	AR	SSA	SPL	EX	AR	SSA	SPL
All	0.67	0.65	0.64	0.41	0.87	0.70	0.84	0.74	0.80	0.65	0.82	0.74
Standing	0.94	0.94	0.92	0.89	0.98	0.94	0.97	0.98	0.95	0.94	0.97	0.96
Elevator-up	0.94	0.94	0.93	0.92	0.95	0.95	0.95	0.95	0.93	0.94	0.94	0.93
Walking-forward	0.87	0.87	0.89	0.70	0.97	0.89	0.96	0.88	0.95	0.87	0.97	0.91
Sitting	0.98	0.95	0.94	0.96	0.99	0.96	0.98	0.99	0.98	0.96	0.99	0.99
Walking-downstairs	0.95	0.93	0.93	0.90	0.99	0.96	0.98	0.95	0.98	0.93	0.98	0.96
Sleeping	1.00	0.98	0.99	1.00	1.00	0.98	1.00	1.00	1.00	0.98	1.00	1.00
Elevator-down	0.94	0.94	0.94	0.91	0.95	0.95	0.95	0.95	0.93	0.94	0.94	0.93
Walking-upstairs	0.94	0.95	0.93	0.92	0.98	0.95	0.98	0.96	0.98	0.95	0.98	0.96
Jumping	0.99	0.99	1.00	0.97	1.00	0.99	1.00	0.99	1.00	0.99	0.97	0.99
Walking-right	0.91	0.90	0.91	0.86	0.97	0.92	0.96	0.92	0.96	0.90	0.97	0.93
Walking-left	0.89	0.91	0.90	0.88	0.97	0.93	0.97	0.93	0.95	0.91	0.97	0.93
Running	0.99	0.99	0.99	0.92	1.00	0.99	1.00	0.97	1.00	1.00	0.95	0.98

данных USC-HAD результат более восприимчив к выбору модели классификации. Для обоих датасетов логистическая регрессия продемонстрировала наименьшее качество, SVM и случайный лес показали почти одинаковое качество.

Для датасета USC-HAD модель с использованием аппроксимации сплайнами показала сравнимое с другими методами качество.

В Табл. 6.3 и Табл. 6.4 представлены результаты классификации (6.4) для каждого класса в отдельности. Первая строка в обеих таблицах демонстрирует точность по всем классам для каждой модели и процедуры генерации признаков. Следующие строки соответствуют бинарным точностям по каждому из классов. Для данных WISDM лучшее качество имеют наименее активные классы, такие как Standing и Sitting. Для USC-HAD заметного выделения качества для определенных классов не наблюдается.

Также был проведён эксперимент с использованием объединённого множества всех 193 сгенерированных признаков. Результаты представлены на Рис. 6.5. Соответствие между номера классов и видами активности приведено в Табл. 6.2. Объединение признаков для обучения одной модели позволило увеличить качество. Для данных WISDM все точности классификации по классам больше 97%, а для USC-HAD выше 93%.

Заключение

Основные результаты диссертационной работы заключаются в следующем.

В главе ?? рассматривается задача декодирования временных рядов. Вводится понятие предсказательной модели. Приведен обзор методов прогнозирования временных рядов. Рассматриваются линейные и нелинейные методы снижения размерности пространства. Описываются методы, работающие с тензорными и многомодальными данными.

В главе ?? вводится понятие скрытого пространства и процедуры согласования. Приведено доказательство корректности линейных методов проекции в скрытое пространство. *Дописать.*

В главе ?? рассматривается задача выбора признаков при декодирования сигналов. Задача выбора признаков ставится как задача дискретной оптимизации. Алгоритм выбора признаков с помощью квадратичного программирования рассматривается как решение релаксированной оптимизационной задачи. Приводятся обобщения процедуры выбора признаков для случая векторной целевой переменной. Предлагаются алгоритмы с симметричным и несимметричным учетом значимости целевых переменных, а также минимаксная постановка задачи.

В главе ?? процедура выбора признаков применяется к выбору параметров для оптимизации нелинейных моделей. Предлагается модификация метода Ньютона для увеличении стабильности процедуры оптимизации. На каждом шаге алгоритма выбирается подмножество активных параметров для оптимизации алгоритмом выбора признаков с помощью квадратичного программирования.

В главе ?? ставится задача выбора оптимальной метрики в пространстве временных рядов. Приводится алгоритм кластеризации, использующий метрику Махalanобиса с обучаемой матрицей для вычисления расстояния между объектами. Для нахождения соответствия между временными рядами приводится алгоритм классификации с процедурой динамического выравнивания времен-

ных рядов.

В главе ?? рассматривается задача порождения признакового пространства при решении задачи классификации временных рядов. Признаковое пространство порождается с помощью метамоделей временных рядов. В качестве метамоделей используются авторегрессионная модель, метод анализа сингулярного спектра, аппроксимация сплайнами.

Список основных обозначений

$\mathbf{s} = (s_1, \dots, s_T)$ – временной ряд длины T

$\mathcal{S} = \{\mathbf{s}_i\}_{i=1}^m$ – множество из m временных рядов

$\mathcal{S}_t = \{s_{it}\}_{i=1}^m$ – временное представление множества временных рядов

$\mathcal{S}_{t-h} = (\mathcal{S}_{t-h}, \dots, \mathcal{S}_t)$ – предыстория множества временных рядов длины h в момент времени t

$\mathcal{S}_{t,p} = (\mathcal{S}_{t+1}, \dots, \mathcal{S}_{t+p})$ – горизонт прогнозирования множества временных рядов длины p в момент времени t

\mathbf{g} – порождающая функция

$\mathbf{X}_{t,-h} = \mathbf{g}(\mathcal{S}_{t,-h})$ – признаковое представление предыстории

$\mathbf{Y}_{t,p} = \mathbf{g}(\mathcal{S}_{t,p})$ – признаковое представление горизонта прогнозирования

$f_{\text{AR}} : \mathbf{Y}_{t,-h} \rightarrow \mathbf{Y}_{t,p}$ – авторегрессионная модель

$f_{\text{R}} : \mathbf{X}_{t,-h} \rightarrow \mathbf{Y}_{t,p}$ – регрессионная модель

$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]^{\top} = [\boldsymbol{\chi}_1, \dots, \boldsymbol{\chi}_n]$ – матрица независимой переменной

$\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_m]^{\top} = [\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_r]$ – матрица целевой переменной

Список иллюстраций

1.1	Модельный пример работы алгоритмов PCA и PLS	19
1.2	Модельный пример работы алгоритмов PCA иCCA	20
2.1	Схема построения моделей декодирования	24
2.2	Примеры алгоритмов, работающих по схеме 2.1	31
2.3	Сигналы мозга (левый график) и 3D координаты руки (правый график)	38
2.4	Прогноз потребления электроэнергии алгоритмом PLS при размерности латентного пространства $l=14$	39
2.5	Зависимость ошибки от размерности латентного пространства для данных потребления электроэнергии	40
2.6	Зависимость ошибки от размерности латентного пространства для данных ECoG	41
2.7	Прогноз движения руки по данным ECoG алгоритмом PLS при размерности латентного пространства $l = 5$	42
2.8	Зашумленные изображения из набора данных MNIST	42
2.9	Набор данных MNIST, в котором каждое изображение разделено пополам	43
2.10	Пример реконструкции правой части изображения по левой для рассматриваемых моделей	45
3.1	Значимости признаков \mathbf{z}_x и целевых векторов \mathbf{z}_y в зависимости от α_3 для рассмотренного примера	52
3.2	Матрицы корреляций для матрицы плана \mathbf{X} и целевой матрицы \mathbf{Y} для данных ECoG	57
3.3	Значимости целевых векторов \mathbf{z}_y в зависимости от α_3 для алгоритма SymImp QPFS	59

3.4	Сравнение предложенных алгоритмов выбора признаков для данных ECoG при прогнозировании $k = 30$ отсчётов времени	60
3.5	Ошибка sRMSE на тестовой выборке для модели PLS	62
3.6	Диаграммы размаха значений sRMSE на тестовой выборке для моделей Lasso, Elastic, LinReg+QPFS, PLS, PLS+QPFS	62
4.1	Поверхность функции ошибки для логистической регрессии	69
4.2	Релевантность параметров для логистической регрессии	69
4.3	Поверхность функции ошибки для нейронной сети	70
4.4	Релевантность параметров первого слоя для модели нейронной сети	70
4.5	Оптимизационный процесс предложенного алгоритма QPFS+Ньютон для модели логистической регрессии	71
4.6	Множество активных параметров на протяжении оптимизационного процесса	71
5.1	Двумерный пример работы алгоритма LMNN	83
5.2	Истинное распределение двумерных модельных данных	84
5.3	Результат кластеризации модельных данных алгоритмом k -средних	85
5.4	Результат кластеризации модельных данных алгоритмом адаптивного метрического обучения	86
5.5	Примеры центроидов синтетических временных рядов	87
5.6	Примеры центроидов временных рядов акселерометра	88
5.7	Примеры временных рядов акселерометра	89
5.8	Выравненные временные ряды акселерометра	89
5.9	Ошибка классификации метрического алгоритма в зависимости от размерности пространства и количества используемых ближайших соседей	90
6.1	Пример аппроксимации временного ряда авторегрессионной моделью с $n = 20$	94

6.2 Пример аппроксимации временного ряда с помощью сплайнов третьего порядка	95
6.3 Примеры временных рядов акселлерометра для каждой оси	98
6.4 Мультиклассовая точность классификации для различных по- рожденных признаковых пространств	100
6.5 Поклассовая точность классификации временных рядов акселе- рометра	101

Список таблиц

1.1	Средняя квадратичная ошибка на модельном примере для алгоритмов линейной регрессии, PCA, PLS, CCA	20
2.1	Точность классификации линейного SVM для алгоритмов Deep CCA и CCA	41
2.2	Квадратичная ошибка для нелинейных моделей в задаче восстановления правой части изображения по левой	44
3.1	Обзор предлагаемых обобщений многомерного QPFS алгоритма .	56
3.2	Стабильность предложенных алгоритмов выбора признаков . .	61
4.1	Средняя квадратичная ошибка на тренировочной и тестовой выборках для модели нелинейной регрессии	72
4.2	Среднее значение кросс-энтропии на тренировочной и тестовой выборках для модели логистической регрессии	73
5.1	Результаты кластеризации на множестве датасетов для методов k -средних и AML	86
5.2	Матрицы несоответствий для Евклидовой метрики и метрики Махalanобиса, построенные для временных рядов акселерометра .	90
5.3	Прирост точности классификации при использовании адекватной оценки матрицы трансформаций	91
6.1	Примеры экспертных порождающих функций	93
6.2	Распределение объектов по классам для временных рядов акселерометра	99
6.3	Бинарная точность классификации для данных WISDM с использованием рассматриваемых алгоритмов: EX – Expert, AR – Auto-Reg, SSA and SPL for Splines	102

6.4 Бинарная точность классификации для данных USC-HAD с ис- пользованием рассматриваемых алгоритмов: EX — Expert, AR — Auto-Reg, SSA and SPL for Splines	102
---	-----

Литература

1. Anastasia Motrenko and Vadim Strijov. Multi-way feature selection for ecog-based brain-computer interface. *Expert Systems with Applications*, 2018.
2. Hyonho Chun and Sündüz Keleş. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(1):3–25, 2010.
3. Tahir Mehmood, Kristian Hovde Liland, Lars Snipen, and Solve Sæbø. A review of variable selection methods in partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 118:62–69, 2012.
4. Alexandr Katrutsa and Vadim Strijov. Comprehensive study of feature selection methods to solve multicollinearity problem according to evaluation criteria. *Expert Systems with Applications*, 76:1–11, 2017.
5. Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94, 2017.
6. Herman Wold. Path models with latent variables: The nipals approach. In *Quantitative sociology*, pages 307–357. Elsevier, 1975.
7. Roman Rosipal and Nicole Kramer. Overview and recent advances in partial least squares. In *International Statistical and Optimization Perspectives Workshop Subspace, Latent Structure and Feature Selection*, pages 34–51. Springer, 2005.
8. Andrey Eliseyev, Vincent Auboironx, Thomas Costecalde, Lilia Langar, Guillaume Charvet, Corinne Mestais, Tetiana Aksenova, and Alim-Louis Benabid. Recursive exponentially weighted n-way partial least squares regression with recursive-validation of hyper-parameters in brain-computer interface applications. *Scientific reports*, 7(1):1–15, 2017.

9. Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.
10. Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics*, pages 162–190. Springer, 1992.
11. A. M. Katrutsa and V. V. Strijov. Stress test procedure for feature selection algorithms. *Chemometrics and Intelligent Laboratory Systems*, 142:172–183, 2015.
12. Irene Rodriguez-Lujan, Ramon Huerta, Charles Elkan, and Carlos Santa Cruz. Quadratic programming feature selection. *Journal of Machine Learning Research*, 11(Apr):1491–1516, 2010.
13. Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.
14. Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2593–2601, 2017.
15. Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216, 2007.
16. Brian Kulis et al. Metric learning: A survey. *Foundations and trends in machine learning*, 5(4):287–364, 2012.
17. Liu Yang and Rong Jin. Distance metric learning: A comprehensive survey. *Michigan State Universiy*, 2(2):4, 2006.
18. Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(2), 2009.
19. Jonathan R Wolpaw, Niels Birbaumer, William J Heetderks, Dennis J McFarland, P Hunter Peckham, Gerwin Schalk, Emanuel Donchin, Louis A

- Quatrano, Charles J Robinson, Theresa M Vaughan, et al. Brain-computer interface technology: a review of the first international meeting. *IEEE transactions on rehabilitation engineering*, 8(2):164–173, 2000.
20. Brendan Z Allison, Elizabeth Winter Wolpaw, and Jonathan R Wolpaw. Brain–computer interface systems: progress and prospects. *Expert review of medical devices*, 4(4):463–474, 2007.
 21. Sebastian Nagel and Martin Spüler. Modelling the brain response to arbitrary visual stimulation patterns for a flexible high-speed brain-computer interface. *PloS one*, 13(10):e0206107, 2018.
 22. Xiang Zhang, Lina Yao, Xianzhi Wang, Jessica JM Monaghan, David Mcalpine, and Yu Zhang. A survey on deep learning-based non-invasive brain signals: recent advances and new frontiers. *Journal of Neural Engineering*, 2020.
 23. Antonio Maria Chiarelli, Pierpaolo Croce, Arcangelo Merla, and Filippo Zappasodi. Deep learning for hybrid eeg-fnirs brain–computer interface: application to motor imagery classification. *Journal of neural engineering*, 15(3):036028, 2018.
 24. Andrey Eliseyev and Tetiana Aksanova. Recursive n-way partial least squares for brain-computer interface. *PloS one*, 8(7):e69962, 2013.
 25. Andrey Eliseyev, Cecile Moro, Thomas Costecalde, Napoleon Torres, Sadok Gharbi, Corinne Mestais, Alim Louis Benabid, and Tatiana Aksanova. Iterative n-way partial least squares for a binary self-paced brain–computer interface in freely moving animals. *Journal of neural engineering*, 8(4):046012, 2011.
 26. P. B. Исаченко. Метрическое обучение в задачах мультиклассовой классификации временных рядов. In *Ломоносов-2016*, pages 129–131, 2016.
 27. R. G. Neychev, A. P. Motrenko, R. V. Isachenko, A. S. Inyakin, and V. V. Strijov. Multimodel forecasting multiscale time series in internet of things. In *Intelligent Data Processing*, pages 130–131, 2016.

28. P. B. Исаченко, И. Н. Жариков, and A. M. Бочкарев. Локальные модели для классификации объектов сложной структуры. In *Математические методы распознавания образов*, volume 18, pages 26–27, 2017.
29. R. V. Isachenko and V. V. Strijov. Dimensionality reduction for multicorrelated signal decoding with projections to latent space. In *Intelligent Data Processing*, pages 86–87, 2018.
30. Р. В. Исаченко and В. В. Стрижов. Снижение размерности в задаче декодирования временных рядов. In *Intelligent Data Processing*, pages 31–32, 2020.
31. Р. В. Исаченко and А. М. Катруца. Метрическое обучение и снижение размерности пространства в задачах кластеризации. *Машинное обучение и анализ данных*, 2(1):17–25, 2016.
32. Роман Владимирович Исаченко and Вадим Викторович Стрижов. Метрическое обучение в задачах мультиклассовой классификации временных рядов. *Информатика и её применения*, 10(2):48–57, 2016.
33. Roman Isachenko, Ilya Zharikov, Artem Bochkarev, and Vadim Strijov. Feature generation for physical activity classification. *Artificial Intelligence and Decision Making*, (3):20–27, 2018.
34. R. V. Isachenko and V. V. Strijov. Quadratic programming optimization with feature selection for nonlinear models. *Lobachevskii Journal of Mathematics*, 39(9):1179–1187, 2018.
35. R. V. Isachenko, M. V. Vladimirova, and V. V. Strijov. Dimensionality reduction for time series decoding and forecasting problems. *DEStech Transactions on Computer Science and Engineering*, (optim), 2018.
36. Ф. Р. Яушев, Р. В. Исаченко, and В. В. Стрижов. Модели согласования скрытого пространства в задаче прогнозирования. *Системы и средства информатики*, 31(1), 2021.
37. George EP Box, Gwilym M Jenkins, and Gregory C Reinsel. *Time series analysis: forecasting and control*, volume 734. John Wiley & Sons, 2011.

38. Keith W Hipel and A Ian McLeod. *Time series modelling of water resources and environmental systems*. Elsevier, 1994.
39. John H Cochrane. Time series for macroeconomics and finance. *Manuscript, University of Chicago*, pages 1–136, 2005.
40. John W Galbraith, Victoria Zinde-Walsh, et al. Autoregression-based estimators for arfima models. Technical report, CIRANO, 2001.
41. Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.
42. Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer, 1997.
43. Svante Wold, Arnold Ruhe, Herman Wold, and WJ Dunn, III. The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3):735–743, 1984.
44. Herman Wold and Jean-Luc Bertholet. The pls (partial least squares) approach to multidimensional contingency tables. *Metron*, 40(1-2):303–326, 1982.
45. Paul Geladi and Bruce R Kowalski. Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17, 1986.
46. Paul Geladi. Notes on the history and nature of partial least squares (pls) modelling. *Journal of Chemometrics*, 2(4):231–246, 1988.
47. Sijmen De Jong. Simpls: an alternative approach to partial least squares regression. *Chemometrics and intelligent laboratory systems*, 18(3):251–263, 1993.
48. V Esposito Vinzi, Wynne W Chin, Jörg Henseler, Huiwen Wang, et al. *Handbook of partial least squares*, volume 201. Springer, 2010.
49. Richard G Brereton and Gavin R Lloyd. Partial least squares discriminant analysis: taking the magic away. *Journal of Chemometrics*, 28(4):213–225, 2014.

50. Roman Rosipal. Nonlinear partial least squares an overview. In *Chemoinformatics and advanced machine learning perspectives: complex computational methods and collaborative techniques*, pages 169–189. IGI Global, 2011.
51. Theodore Wilbur Anderson. An introduction to multivariate statistical analysis. Technical report, Wiley New York, 1962.
52. Shotaro Akaho. A kernel method for canonical correlation analysis. *arXiv preprint cs/0609071*, 2006.
53. Thomas Melzer, Michael Reiter, and Horst Bischof. Nonlinear feature extraction using generalized canonical correlation analysis. In *International Conference on Artificial Neural Networks*, pages 353–360. Springer, 2001.
54. Francis R Bach and Michael I Jordan. Kernel independent component analysis. *Journal of machine learning research*, 3(Jul):1–48, 2002.
55. David R Hardoon, Sandor Szédmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.
56. David R Hardoon, Janaina Mourao-Miranda, Michael Brammer, and John Shawe-Taylor. Unsupervised analysis of fmri data using kernel canonical correlation. *NeuroImage*, 37(4):1250–1259, 2007.
57. Alexei Vinokourov, Nello Cristianini, and John Shawe-Taylor. Inferring a semantic representation of text via cross-language correlation analysis. In *Advances in neural information processing systems*, pages 1497–1504, 2003.
58. Luca Montanarella, Maria Rosa Bassani, and Olivier Bréas. Chemometric classification of some european wines using pyrolysis mass spectrometry. *Rapid Communications in Mass Spectrometry*, 9(15):1589–1593, 1995.
59. Jean-Philippe Vert and Minoru Kanehisa. Graph-driven feature extraction from microarray data using diffusion kernels and kernel cca. In *Advances in neural information processing systems*, pages 1449–1405, 2003.

60. Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: Hlt*, pages 771–779, 2008.
61. Paramveer Dhillon, Dean P Foster, and Lyle H Ungar. Multi-view learning of word embeddings via cca. In *Advances in neural information processing systems*, pages 199–207, 2011.
62. K Choukri and G Chollet. Adaptation of automatic speech recognizers to new speakers using canonical correlation analysis techniques. *Computer Speech & Language*, 1(2):95–107, 1986.
63. Frank Rudzicz. Adaptive kernel canonical correlation analysis for estimation of task dynamics from acoustics. In *ICASSP*, pages 4198–4201, 2010.
64. Tae-Kyun Kim, Shu-Fai Wong, and Roberto Cipolla. Tensor canonical correlation analysis for action classification. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
65. Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255. PMLR, 2013.
66. Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *International conference on machine learning*, pages 1083–1092, 2015.
67. Xiaobin Chang, Tao Xiang, and Timothy M Hospedales. Scalable and effective deep cca via soft decorrelation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1488–1497, 2018.
68. Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.
69. Qibin Zhao, Cesar F Caiafa, Danilo P Mandic, Zenas C Chao, Yasuo Nagasaka, Naotaka Fujii, Liqing Zhang, and Andrzej Cichocki. Higher order partial least

- squares (hopls): a generalized multilinear regression method. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1660–1673, 2012.
70. Andrey Eliseyev and Tetiana Aksenova. Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ecog) recording. *PloS one*, 11(5):e0154878, 2016.
 71. Einat Kidron, Yoav Y Schechner, and Michael Elad. Pixels that sound. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 88–95. IEEE, 2005.
 72. Kamalika Chaudhuri, Sham M Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th annual international conference on machine learning*, pages 129–136, 2009.
 73. Raman Arora and Karen Livescu. Kernel cca for multi-view learning of acoustic features using articulatory measurements. In *Symposium on Machine Learning in Speech and Language Processing*, 2012.
 74. Richard Socher and Li Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 966–973. IEEE, 2010.
 75. Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.
 76. Sarath Chandar AP, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha. An autoencoder approach to learning bilingual word representations. In *Advances in neural information processing systems*, pages 1853–1861, 2014.
 77. Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, 2014.

78. Jonathan Masci, Michael M Bronstein, Alexander M Bronstein, and Jürgen Schmidhuber. Multimodal similarity-preserving hashing. *IEEE transactions on pattern analysis and machine intelligence*, 36(4):824–830, 2013.
79. Janarthanan Rajendran, Mitesh M Khapra, Sarath Chandar, and Balaraman Ravindran. Bridge correlational neural networks for multilingual multimodal representation learning. *arXiv preprint arXiv:1510.03519*, 2015.
80. Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized multi-view spectral clustering. In *Advances in neural information processing systems*, pages 1413–1421, 2011.
81. Abhishek Sharma, Abhishek Kumar, Hal Daume, and David W Jacobs. Generalized multiview analysis: A discriminative latent space. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2160–2167. IEEE, 2012.
82. Paul Geladi. Notes on the history and nature of partial least squares (PLS) modelling. *Journal of Chemometrics*, 2(January):231–246, 1988.
83. Kentaro Shimoda, Yasuo Nagasaka, Zenas C Chao, and Naotaka Fujii. Decoding continuous three-dimensional hand trajectories from epidural electrocorticographic signals in japanese macaques. *Journal of neural engineering*, 9(3):036015, 2012.
84. Zenas C Chao, Yasuo Nagasaka, and Naotaka Fujii. Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkey. *Frontiers in neuroengineering*, 3:3, 2010.
85. Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017.
86. Chris Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proc. SIAM Data Mining Conf*, 2005.
87. John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

88. Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.
89. The wisdm dataset. <http://www.cis.fordham.edu/wisd़/dataset.php>.
90. M.E. Karasikov and V.V. Strijov. Feature-based time-series classification. *Intelligence*, 24(1):164–181, 2016.
91. M.P. Kuznetsov and N.P. Ivkin. Time series classification algorithm using combined feature description. *Machine Learning and Data Analysis*, 1(11):1471–1483, 2015.
92. Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.
93. Yu P Lukashin. Adaptive methods of short-term forecasting of time series. *M.: Finance and statistics*, 2003.
94. Hossein Hassani. Singular spectrum analysis: methodology and comparison. *Journal of Data Science*, 5(2):239–257, 2007.
95. Carl De Boor. *A practical guide to splines*, volume 27. Springer-Verlag, 1978.
96. The usc human activity dataset. <http://www-scf.usc.edu/~mizhang/datasets.html>.